

N69-36473

NASA CR-66819

FINAL REPORT

SIMULATED TRAJECTORIES ERROR ANALYSIS PROGRAM

VOLUME II: ANALYTICAL MANUAL

By Gentry Lee, Ralph Falce, Dr. Doyle Vogt,  
Shearon Pearson, and Eva Demlow

Distribution of this report is provided in the  
interest of information exchange. Responsibility  
for the contents resides in the author or  
organization that prepared it.

Prepared under Contract No. NAS1-8745 by  
MARTIN MARIETTA CORPORATION  
Denver, Colorado

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
Langley Research Center  
Hampton, Virginia





## FOREWORD

This document represents Volume II of the final report on NASA Langley Contract NAS1-8745 entitled "Simulated Trajectories Error Analysis Program". The report is prepared in two volumes:

Volume I - User's Manual;

Volume II - Analytical Manual.





## CONTENTS

	Page
FOREWORD . . . . .	iii
SUMMARY . . . . .	1
I. INTRODUCTION . . . . .	2
A. Trajectory Mode . . . . .	2
B. Targeting Mode . . . . .	3
C. Error Analysis Mode . . . . .	4
D. Simulation Mode . . . . .	6
II. SYMBOLS . . . . .	7
III. TRAJECTORY MODE - ANALYSIS . . . . .	9
IV. TARGETING MODE - ANALYSIS . . . . .	18
A. Statement of Problem . . . . .	18
B. General Method of Solution . . . . .	19
C. General Definition of Target Conditions . . . . .	24
D. Generation of Patched Conic Injection Conditions . . . . .	25
E. Auxiliary Computations in Targeting . . . . .	32
F. Specifics of Targeting Scheme . . . . .	39
V. ERROR ANALYSIS MODE - ANALYSIS . . . . .	41
A. SUMMARY . . . . .	41
B. Recursive Estimation Algorithm . . . . .	43
C. State Transition Matrices . . . . .	47
D. Measurement Processing . . . . .	53
E. Midcourse Guidance Equations . . . . .	66
F. Eigenvector and Prediction Events . . . . .	82
G. State Vector Augmentation . . . . .	86
VI. SIMULATION MODE - ANALYSIS . . . . .	96
A. The Basic Cycle . . . . .	96
B. Guidance Event . . . . .	102
C. Other Events . . . . .	107
D. Divergence and Other Problems . . . . .	108
VII. NUMERICAL RESULTS . . . . .	115
A. Trajectory Mode . . . . .	115
B. Targeting Mode . . . . .	139
C. Error Analysis Mode . . . . .	154
D. Simulation Mode . . . . .	205
VIII. REFERENCES . . . . .	233
	and
	234
<b>Figure</b>	
1 Impact Plane Parameters . . . . .	24
2 Spherical Trigonometry for Closest Approach Conversion . . . . .	34

3	Projection of Orbital Plane and Normal on Impact Plane . . . . .	37
4	Outer Targeting . . . . .	38
5	Station Position . . . . .	56
6	Accuracy Analysis in Detail, Plot of $\Delta R$ vs Trajectory Time . . . . .	122
7	Accuracy Analysis in Detail, Plot of $\Delta V$ vs Trajectory Time . . . . .	123
8	Position of Planets during Test Trajectory . . . . .	126
9	Ephemeris Analysis in Detail, Plot of $\Delta R$ vs Trajectory Time . . . . .	133
10	Ephemeris Analysis in Detail, Plot of $\Delta V$ vs Trajectory Time . . . . .	135
11	Processed Navigation Uncertainties from Base Error Analysis Run . . . . .	159
12	Navigation Uncertainties from Base Error Analysis Run . . . . .	163
13	Comparison of Late Navigation Uncertainties with and without 3rd Midcourse . . . . .	172
14	Navigation Uncertainties for Error Analysis Run . . . . .	176
15	Navigation Uncertainties for Auxiliary Error Analysis Run . . . . .	188
16	Navigation Uncertainties for Error Analysis Run in Mars Orbit . . . . .	194

Table

1	Mars-Viking Run, Summary of Accuracy Analysis . . . . .	117
2	Jupiter Run, Summary of Accuracy Analysis . . . . .	117
3	Mars-Viking Run, Detailed Accuracy Analysis for 6, 30, and 60 Days . . . . .	119
4	Mars-Viking Run, Detailed Accuracy Analysis for 90, 120, and 150 Days . . . . .	120
5	Mars-Viking Run, Detailed Accuracy Analysis for 180, 194, and 199 Days . . . . .	121
6	Mars-Viking Run, Summary of Ephemeris Analysis . . . . .	125
7	Mars-Viking Run, Detailed Ephemeris Analysis for 6, 60, and 120 Days . . . . .	131
8	Mars-Viking Run, Detailed Ephemeris Analysis for 180, 194, and 199 Days . . . . .	132
9	Mars-Viking Run, Summary of Targeted Accuracy Analysis . . . . .	137
10	Mars-Viking Run, Detailed Targeted Accuracy Analysis for 20, 100, and 197 Days . . . . .	138
11	Summary of Target Options . . . . .	141
12	Interplanetary Trajectory Targeting Program . . . . .	142
13	Actual Targeting Procedures . . . . .	143

14	Distribution of CDC 6500 Computer Time in Types of Integrations . . . . .	146
15	Comparison of Target Options 5 and 6 . . . . .	146
16	Family of Solutions to Closest Approach Problem . .	147
17	Comparison of Target Options 3 and 4 . . . . .	148
18	Demonstration of Outer Targeting . . . . .	150
19	Demonstration of Bad-Step Check . . . . .	151
20	Analysis of Minimum Velocity Midcourse Correction .	152
21	Targeting of Swingby Mission . . . . .	153
22	Conditions for Base Error Analysis Run . . . . .	155
23	Navigation Uncertainties for Base Error Analysis Run . . . . .	157
24	Navigation Uncertainties at Closest Approach for Base Run . . . . .	161
25	Guidance Correction Information for Base Error Analysis Run . . . . .	166
26	Target Condition Uncertainties for Base Error Analysis Run . . . . .	169
27	Late Navigation Uncertainties for Base Run Without 3rd Midcourse Correction . . . . .	171
28	Early Navigation Uncertainties for Higher Accuracy Base Run . . . . .	174
29	Late Navigation Uncertainties for High Accuracy Run Without 3rd Midcourse Correction . . . . .	175
30	Scattered Navigation Uncertainties for Poor Injection Run Without 3rd Midcourse Correction But with High Accuracy . . . . .	177
31	Guidance Correction Information for Low Dynamic Noise Run . . . . .	179
32	Guidance Correction Information for High Dynamic Noise Run . . . . .	179
33	Different Measurement Schedule for Auxiliary Error Analysis Run . . . . .	181
34	Early Navigation Uncertainties for Auxiliary Error Analysis Run . . . . .	182
35	Late Navigation Uncertainties for Auxiliary Error Analysis Run . . . . .	183
36	Target Condition Uncertainties for Auxiliary Error Analysis Run, Augmented vs Nonaugmented . . . . .	185
37	Early Navigation Uncertainties for Augmented Run .	186
38	Late Navigation Uncertainties for Augmented Run . .	187
39	Variances of Augmented Parameters for Augmented Run . . . . .	190
40	Conditions for Error Analysis Run in Mars Orbit . .	192
41	Navigation Uncertainties for Error Analysis Run in Mars Orbit . . . . .	193
42	Correlation Coefficients for Mars in Orbit Run . .	196

43	Conditions for Venus Error Analysis Run . . . . .	198
44	Comparison of Venus Runs, $\Delta V$ . . . . .	199
45	Comparison of Target Condition Uncertainties for Different 1st Midcourse Times . . . . .	201
46	Comparison of Target Condition Uncertainties after Two Corrections . . . . .	202
47	Comparison of Target Condition Uncertainties after All Corrections . . . . .	203
48	Comparison of Navigation Uncertainties at Closest Approach . . . . .	204
49	Conditions for Simulation Mode Run 1 . . . . .	206
50	Orbit Determination Inaccuracies for Simulation Mode Run 1 . . . . .	208
51	Statistical Consistency Residual Check for Simulation Mode Run 1 . . . . .	211
52	Guidance Correction Information for Simulation Mode Run 1 . . . . .	213
53	Changes in Conditions for Simulation Mode Run 2 . .	215
54	Orbit Determination Inaccuracies for Simulation Mode Run 2 . . . . .	215
55	Statistical Consistency Residual Check for Simulation Mode Run 2 . . . . .	217
56	Guidance Correction Information for Simulation Mode Run 2 . . . . .	219
57	Changes in Conditions for Simulation Mode Run 3 . .	220
58	Orbit Determination Inaccuracies for Simulation Mode Run 3 . . . . .	221
59	Statistical Consistency Residual Check for Simulation Mode Run 3 . . . . .	222
60	Guidance Correction Information for Simulation Mode Run 3 . . . . .	224
61	Change in Conditions for Simulation Mode Runs 4 and 5 . . . . .	225
62	Orbit Determination Inaccuracies for Simulation Mode Run 4 . . . . .	226
63	Estimated Parameter Values for Simulation Mode Run 4 . . . . .	228
64	Orbit Determination Inaccuracies for Simulation Mode Run 5 . . . . .	229
65	Estimated Parameter Values for Simulation Mode Run 5 . . . . .	230
66	State Transition Matrix Comparison for Simulation Mode Runs 4 and 5 . . . . .	232

FINAL REPORT  
SIMULATED TRAJECTORIES ERROR ANALYSIS PROGRAM

VOLUME II: ANALYTICAL MANUAL

By Gentry Lee, Dr. Doyle Vogt, Ralph Falce,  
Shearon Pearson, and Eva Demlow  
Martin Marietta Corporation

SUMMARY

Volume I of this report under NASA-Langley Contract NAS1-8745 contains a user's manual and guide to the operation of the Simulated Trajectories Error Analysis Program (STEAP) developed under the contract. This volume explains the mathematical techniques used in the program and gives typical results, with explanations, that demonstrate the efficacy of the program.

Chapter I contains a general description of STEAP and introduces the operational modes of the program. Chapter II contains the symbols used through the text. In Chapters III thru VI the analytical techniques used for the development of each operational mode of the program are presented in detail. Chapter VII contains a discussion of many test runs made with the program and is intended to demonstrate the multifarious uses for the program. Chapter VII also includes additional information of a general nature that should speed the program user's development of an intuitive feel for the program.

## I. INTRODUCTION

The Simulated Trajectories Error Analysis Program (STEAP) is comprised of four subprograms or operational modes. The first mode, which is used as a subroutine by all the remaining modes, is the trajectory mode. The second mode or subprogram is called the targeting mode. The third operational mode of STEAP has been designated as the error analysis mode. The fourth and final mode in which the program can be exercised is the simulation mode. The following subsections describe each of these four operational modes.

### A. Trajectory Mode

The trajectory mode essentially computes an n-body trajectory using the varicentric or virtual mass technique explained in reference 1. Input to the trajectory mode is some set of initial conditions that may have resulted from the targeting mode, plus parameters necessary to specify the computational operations used by the program. No actual integration is performed by the trajectory mode; the key idea of the virtual mass technique is to build up an n-body trajectory by using sequences of conic sections around a moving effective force center called the virtual mass. At each instantaneous moment along the trajectory, the combined effects of all the gravitational bodies can be viewed as resulting from a fictitious body, of unique magnitude and position, which is called the virtual mass. The computational procedure then assumes that over a small time interval, the motion of the spacecraft can be represented by a two-body conic section arc relative to this virtual mass. The complete trajectory is thus generated by a series of small arcs pieced together in steps while updating the position and magnitude of the effective force center. The main advantage of the virtual mass technique is that numerical integration of the differential equations is not necessary.

The trajectory mode has access to an ephemeris subroutine that calculates, for a given date, the conic section orbital elements of the planets. This ephemeris subroutine using orbital elements gives an adequate representation of the state of the solar system for the computation of interplanetary trajectories. If the user desires, each of the planets can be set in a fixed ellipse referred to some epoch for speedier computation.

Another significant feature of the virtual mass technique for generating n-body trajectories is its flexibility. By varying a simple control parameter related to the time increment over which the virtual mass position and magnitude are assumed fixed, trajectories ranging from a series of relatively few conic section arcs (an approximate solution) to highly accurate orbits can be generated. Thus, by using virtual mass concepts, one can predetermine the accuracy needed for a particular mission analysis study, and hence, vary the computational time needed to generate the trajectory.

### B. Targeting Mode

The targeting mode of STEAP is responsible for the generation of interplanetary mission injection conditions. Specifically, the targeting mode computes the injection position and velocity which, when propagated forward along an n-body trajectory, yields a trajectory satisfying specified targeting conditions.

The mathematical model for the trajectory is set by designating the computational step size to be used by the virtual mass subroutine and the particular gravitational bodies to be included in the integration. The launch data, launch planet, target date, and target planet are also specified. The resulting targeted trajectory then meets these constraints as well as some set of target conditions.

Six options are permitted in specifying these target conditions. The first two of these options, which are generally ancillary to the others, are sometimes useful by themselves. In these options the injection conditions corresponding to a crude patched conic trajectory from the launch planet to the target planet are generated. These conditions are the injection position, velocity, and time consistent with an internally-supplied launch profile originating from Cape Kennedy on the desired date. In the first option, conditions relevant to an n-body trajectory are generated; in the second option conditions useful for exact patched conics are produced.

The four basic targeting options are all built on n-body trajectories resulting from the virtual mass subroutine. The second and third options involve the impact plane parameters  $B \cdot T$  and  $B \cdot R$  and the time of arrival at the target planet sphere of influence  $t_{SI}$ . In the second targeting mode option, injection conditions are generated that, when propagated along an n-body trajectory, satisfy target values  $B \cdot T$  and  $B \cdot R$  to prescribed tolerances while satisfying the time constraint only approximately.

The third option forces the satisfaction of the time constraints as well. In the fourth and fifth options the target parameters are the radius of closest approach  $r_{CA}$ , the inclination (with respect to the target planet equator) at closest approach  $i_{CA}$ , and the time at closest-approach  $t_{CA}$ . In the fourth option these target parameters are met approximately; in the fifth they are satisfied to within specified input tolerances.

The targeting program uses numerical differencing (on the injection velocity components) in an iterative process to generate the desired injection conditions for the four basic targeting options. It therefore requires an initial iterate to start the procedure. To allow versatility in the types of problems handled by the program, alternative methods of specifying these conditions are provided. In the usual problem, no first guess is available and the program merely generates the point-to-point conditions described as the first targeting option. A second option, in which the first guess injection conditions are input to the program, allows for the convenient retargeting of nominal trajectories and refined targeting of partially targeted trajectories.

Thus the targeting mode offers a convenient and flexible tool for the generation of the trajectories needed by an analyst of interplanetary missions and is an integral part of STEAP.

### C. Error Analysis Mode

The error analysis mode is essentially a tool for preflight mission analysis and is primarily concerned with covariance matrix propagation along selected interplanetary trajectories. The output of an error analysis run contains information about orbit determination uncertainties in the neighborhood of the target planet, likely fuel expenditures at guidance corrections, and probabilities of target condition miss resulting from execution errors at simulated midcourse maneuvers.

The options in the error analysis mode are quite numerous. State transition matrices can be computed by any of three techniques--analytical patched conic, analytical virtual mass, or numerical differencing. The measurements being processed in an optimal recursive filter may be range-rate or range and range rate from one or more (up to three) stations on a rotating earth, as well as onboard measurements of apparent planet diameter and three star planet angles. Measurement schedules are completely



arbitrary and measurement accuracies are variable input. Perturbation of the equations of motion by the addition of dynamic or process noise is also permitted in the error analysis mode.

The error analysis mode also has several auxiliary capabilities that allow for special events where additional computations are made. There are three events available to the user--eigenvector, prediction, and guidance--the times of which are specified in the input. At an eigenvector event, the basic computational cycle is interrupted and the covariance matrix of navigation uncertainty is computed. Then the associated eigenvalues and eigenvectors of the position and/or velocity uncertainties are calculated. At a prediction event, the error analysis mode interrupts the basic cycle and computes the predicted navigation uncertainty at some future critical time. When a guidance event occurs, the error analysis mode also interrupts the basic cycle and adds a maneuver execution uncertainty to the propagated covariance matrix. Since no actual guidance corrections are made in the error analysis mode, modeling of the execution uncertainty is achieved by considering ensemble errors resulting from four execution error sources: resolution, proportionality, and two pointing errors. Two separate options are available for computing the midcourse execution uncertainty. The computation of the execution error covariance matrix is dependent upon the choice of guidance policy. Three guidance policy options are available to the program: fixed time of arrival, two variable B plane, and three variable B plane.

In addition to the foregoing options, the treatment of augmented states is possible in the error analysis mode. Exercising the augmented state options in the error analysis mode results in the propagation of uncertainties related to measurement biases and parameter errors of various types, as well as position and velocity. The nonaugmented state vector consists of position and velocity and has a  $6 \times 1$  dimension. Ten augmented state vector options are available ranging from an  $8 \times 1$  state vector to a  $17 \times 1$  state vector. Included as augmentation parameters under the various options are biases in the gravitational parameters of the Sun and target planet, as many as nine station location coordinate biases, and six measurement biases. Also included in several of the augmented state options are three target planet ephemeris biases that model uncertainties in target planet semimajor axis, eccentricity, and inclination.

#### D. Simulation Mode

The simulation mode of STEAP is the most complex operational mode. The primary purpose of the simulation mode is to test the orbit determination process exemplified by the error analysis mode. In the error analysis mode, navigation uncertainties are simply propagated along a nominal trajectory. In contrast, the simulation mode determines the validity of the statistical estimation process by simulating an "actual" trajectory. The results given by the error analysis mode become meaningful only when, in the simulation mode, the orbit determination process converges to the "actual" trajectory.

The computations in the simulation mode are similar to those in the error analysis mode, except for the computation of many additional quantities. Four related trajectories are carried through the simulation mode basic cycle: the "actual" trajectory, the estimated trajectory, the original nominal trajectory (used for guidance), and the updated nominal trajectory.

There are four types of events that can be used in the simulation mode. Three of the events are the same as in the error analysis mode--eigenvector, prediction, and guidance. One additional event that can be exercised in the simulation mode is a quasi-linear filtering event. The eigenvector and prediction events in the simulation mode are the same as they were in the error analysis mode.

When a guidance event occurs in the simulation mode, an "actual" midcourse correction is made based on one of the three possible guidance policies. The resulting execution error statistics are computed and an "actual" correction is then calculated that simulates midcourse engine performance. At a quasi-linear filtering event, the nominal trajectory is updated by taking into account the estimated deviation from the most recent nominal trajectory. The reason for including a quasi-linear filtering event is to be able to compute more accurate state transition matrices and check the effect of updating the nominal trajectory on the occurrence of divergence in the filter.

The available options in the simulation mode are identical with those in the error analysis mode. Measurement accuracies can be varied as well as measurement schedules. The state transition matrix options (choice of three) are available as well as a choice of three guidance policies. The augmented state vectors

treated by the simulation mode gives the user the option of estimating such parameters as planetary ephemerides and masses, measurement biases, and station location errors, as well as the basic six-vector of position and velocity.

## II. SYMBOLS

B·T, B·R impact plane parameters

$\tilde{dX}_k$  actual state vector deviation from most recent nominal trajectory at time  $t_k$

$H_{k+1}$  observation matrix at time  $t_{k+1}$

$K_{k+1}$  Kalman gain matrix at time  $t_{k+1}$

$\bar{P}_{k+1}$  covariance matrix of navigation uncertainty at the time  $t_{k+1}$

$P_{k+1}$  covariance matrix of navigation uncertainty after the measurement has been processed at the time  $t_{k+1}$

$P_o$  initial covariance matrix

$Q_{k+1}$  dynamic noise matrix

$\tilde{Q}_{k+1}$  execution error matrix

$R_{k+1}$  covariance matrix of a zero mean Gaussian white noise corrupting the measurement

$\bar{R}_{k+1}$  actual covariance matrix of a zero mean Gaussian white noise corrupting the measurement

$S_{k+1}$  residual uncertainty matrix

$\bar{X}_o$  initial state vector

$\bar{X}_k$  original nominal state vector

$\tilde{X}_k$  most recent nominal state vector

Greek

$\Gamma$	guidance matrix
$\tilde{\delta X}_k$	estimated state vector deviation from most recent nominal trajectory
$\Delta V$	midcourse correction velocity increment
$\epsilon_{k+1}$	measurement residuals
$\eta$	variation matrix of B·T and B·R
$\mu_j$	gravitational parameter of $j^{\text{th}}$ body
$v_{k+1}$	sampled measurement noise
$\phi_{k+1, k}$	state transition matrix relating the times $t_{k+1}$ and $t_k$
$\omega_{k+1}$	actual unmodeled accelerations

### III. TRAJECTORY MODE - ANALYSIS

The trajectory mode of STEAP computes an n-body trajectory for an infinitesimal spacecraft through the use of the varicentric or virtual mass concept. As explained in detail by Novak in reference 1, the essential idea of virtual mass n-body trajectory computations is that, at any instant of time, the gravitational forces exerted by all the governing bodies can be resolved into one effective vector emanating from a virtual mass whose position and magnitude are uniquely determined. Over small time intervals, therefore, the motion of the spacecraft can be represented as a two-body conic section arc around the moving and varying virtual mass. The computational algorithm of the STEAP trajectory mode uses this concept in determining the n-body spacecraft trajectory.

Novak's original work proved the validity of the virtual mass approach for the restricted three-body problem. The trajectory mode of STEAP extends its applicability to general n-body problems. Modeled in the trajectory mode are the best available mean conic section orbital elements of each of the planets in the solar system plus the Earth's moon. These are available to the trajectory mode through an ephemeris subroutine and permit the determination of realistic interplanetary trajectories.

The basic concepts of virtual mass n-body trajectory computation are reviewed here for reference. In addition, the computational algorithm at each interval along the trajectory is presented, step by step, just as it appears in the trajectory mode of STEAP. For more details concerning the underlying concepts, see reference 1.

Consider the vector differential equations for the motion of an infinitesimal spacecraft under the influence of n attracting bodies to be given by:

$$\ddot{\vec{r}}_s = \sum_{i=1}^n \mu_i \frac{(\vec{r}_i - \vec{r}_s)}{|\vec{r}_i - \vec{r}_s|^3} \quad (1)$$

where  $\vec{r}_s$  is the position vector of the spacecraft in some reference coordinate system,  $\mu_i$  is the gravitational attraction of the  $i^{\text{th}}$  governing body, and  $\vec{r}_i$  is the position vector of the  $i^{\text{th}}$  body in the same reference system. It is easy to show that an equivalent set of equations can be written as:

$$\ddot{\vec{r}}_s = \ddot{\vec{M}} - \ddot{\vec{r}}_s M_s = \frac{\mu_v (\vec{r}_v - \vec{r}_s)}{|\vec{r}_v - \vec{r}_s|^3} \quad (2)$$

where the quantities  $M$ ,  $M_s$ ,  $\mu_v$ , and  $\vec{r}_v$  are defined by:

$$\vec{M} = \sum_{i=1}^n \frac{\mu_i \vec{r}_i}{|\vec{r}_i - \vec{r}_s|^3} \quad (3)$$

$$M_s = \sum_{i=1}^n \frac{\mu_i}{|\vec{r}_i - \vec{r}_s|^3} \quad (4)$$

$$\vec{r}_v = \frac{\vec{M}}{M_s} \quad (5)$$

$$\mu_v = |\vec{r}_v - \vec{r}_s|^3 M_s \quad (6)$$

The final form of equation (2), which is easily recognized as the differential equation for two-body motion, suggests the essential idea of virtual mass computation for n-body trajectories. Over intervals where  $\mu_v$  and  $\vec{r}_v$  can be treated as constants, the motion is two-body with respect to this magnitude and position of the virtual mass. The computation of the n-body orbit in the trajectory mode results from piecing together two-body arcs around varying magnitudes and locations for the virtual mass. The way in which each two-body arc is calculated is discussed in the computational algorithm to follow.

A close inspection of the above equations demonstrates that the location and magnitude of the virtual mass has the desired limiting properties. When the spacecraft is within the influence of one dominant body, the virtual mass position and magnitude approximate those of the dominant body. In a transition region, two or more bodies may contribute significantly to the location and magnitude of the virtual mass.

The computational scheme used within the trajectory mode of STEAP is presented in the following paragraphs. Emphasis is placed on the procedure used for determining each individual two-body arc in the sequence. The decision concerning the length of the interval

is made before entering the computational algorithm and is based on a fixed true anomaly passage with respect to the virtual mass. Thus, when the spacecraft is near the virtual mass (near a planet for interplanetary applications), smaller steps are taken to ensure the accuracy of the computation. Over the heliocentric portions of an interplanetary flight, when the spacecraft is far from the virtual mass location and the trajectory is essentially a heliocentric ellipse, larger computational intervals are automatically used.

Within each computing interval, the motion of the virtual mass is assumed to be constant velocity with a constant mass magnitude. Two approaches to determining this constant velocity and mass magnitude were analyzed by Novak (ref. 1). One is called the iterative method and the other noniterative. The noniterative computation uses the values for the virtual mass velocity and mass magnitude at the beginning of the time step for the entire computation interval. Then, at the beginning of the new time step, new values are calculated and assumed consistent with the new position of the spacecraft. This method results in position discontinuities in the virtual mass trajectory since the initial values, rather than any computed mean values, are used over the step. The spacecraft trajectory itself is still continuous, but it is being based on a discontinuous virtual mass trajectory.

In the iterative method "average" values for the virtual mass velocity and mass magnitude are used over the computing interval. The values of  $\vec{r}_v$  and  $\mu_v$  at the end of the interval are initially estimated and then iteratively improved to force consistency between the virtual mass and spacecraft trajectories. The iterative method is used in STEAP; its computational algorithm is presented here, step by step.

- 1) At the beginning of the  $n^{\text{th}}$  time interval the acceleration terms from the previous time interval are calculated as

$$\begin{aligned} \ddot{\vec{r}}_{v \text{ av}}^n &= \frac{\vec{r}_{vE}^{n-1} - \vec{r}_{vB}^{n-1} - \dot{\vec{r}}_{vB}^{n-1} (\Delta t^{n-1})}{(\Delta t^{n-1})^2} \\ \ddot{\mu}_{v \text{ av}}^n &= \frac{\mu_{vE}^{n-1} - \mu_{vB}^{n-1} - \dot{\mu}_{vB}^{n-1} (\Delta t^{n-1})}{(\Delta t^{n-1})^2} \end{aligned} \quad (7)$$

At the first time interval  $\ddot{\vec{r}}_{v \text{ av}} = 0$  and  $\ddot{\mu}_{v \text{ av}} = 0$ .

- 2) Assuming a second order variation with time, an initial guess for the final position and magnitude of the virtual mass is made by using the equations,

$$\begin{aligned}\vec{r}_{v_E} &= \vec{r}_{v_B} + \dot{\vec{r}}_{v_B} (\Delta t) + \ddot{\vec{r}}_{v_{av}} (\Delta t)^2 \\ \mu_{v_E} &= \mu_{v_B} + \dot{\mu}_{v_B} (\Delta t) + \ddot{\mu}_{v_{av}} (\Delta t)^2\end{aligned}\tag{8}$$

The superscripts  $n$  have been dropped for convenience.

All the succeeding equations are for the  $n^{\text{th}}$  time interval unless otherwise specified. The subscripts B and E refer to the beginning and end of the computational interval.

- 3) Using the assumed position and mass magnitude of the virtual mass at the end of the time interval, the assumed average velocity and mass magnitude for the virtual mass over the computational interval may be calculated as

$$\dot{\vec{r}}_{v_{av}} = \frac{\vec{r}_{v_E} - \vec{r}_{v_B}}{\Delta t}\tag{9}$$

$$\mu_{v_{av}} = C_1 \mu_{v_E} + (1 - C_1) \mu_{v_B}$$

where  $C_1$  linearly interpolates the virtual mass magnitude to some value between the initial and final values ( $0 \leq C_1 \leq 1$ ).

The initial velocity of the spacecraft with respect to the virtual mass is now based on this assumed average velocity and is given by

$$\dot{\vec{r}}_{v_{s_B}} = \dot{\vec{r}}_{s_B} - \dot{\vec{r}}_{v_{av}}\tag{10}$$

- 4) The Keplerian vector (represents twice the areal rate) for the computing interval is next computed as

$$\vec{k} = \vec{r}_{v_{s_B}} \times \dot{\vec{r}}_{v_{s_B}}\tag{11}$$



Then the eccentricity vector is determined from

$$\vec{e} = -\frac{\vec{r}_{v s_B}}{r_{v s_B}} - \frac{\vec{k} \times \vec{r}_{v s_B}}{\mu_{v av}} \quad (12)$$

The magnitude of the eccentricity vector,  $\vec{e}$ , represents the eccentricity of the conic section and the orientation of the vector is toward the conic section periapsis.

- 5) The final position and velocity of the spacecraft with respect to the virtual mass is calculated next. An intermediate variable  $\Delta\tau$  is used which must be related to the desired  $\Delta t$  for the interval. The value  $\Delta\tau$  determines the time or true anomaly increment along the conic section arc. Again assuming a second order variation,

$$\Delta\tau = \Delta t + \kappa \Delta t^2 \quad (13)$$

where  $\kappa$  is computed from information about the preceding interval as

$$\kappa = \frac{\Delta\tau - \Delta t}{(\Delta t)^2} \quad (14)$$

The final position  $\vec{r}_{v s_E}$  must lie in the plane of motion defined by  $\vec{r}_{v s_B}$  and  $\dot{\vec{r}}_{v s_B}$ , and hence can be expressed as a linear combination of the two,

$$\vec{r}_{v s_E} = B \left[ \vec{r}_{v s_B} + \Delta\tau \dot{\vec{r}}_{v s_B} \right] \equiv B \vec{\sigma}_{v s_E} \quad (15)$$

where the quantity  $B$  is given by,

$$B = \frac{k^2 / \mu_{v av}}{\vec{e} \cdot \vec{\sigma}_{v s_E} + \sigma_{v s_E}} \quad (16)$$

Then the velocity of the spacecraft with respect to the virtual mass at the end of the interval is

$$\dot{\vec{r}}_{v s_E} = \frac{\vec{k} \times \left( \vec{e} + \frac{\vec{r}_{v s_E}}{r_{v s_E}} \right)}{\left( k^2 / \mu_{v av} \right)} \quad (17)$$

- 6) The final position and velocity of the spacecraft in the reference coordinate system are now computed from

$$\begin{aligned} \vec{r}_{s_E} &= \vec{r}_{v s_E} + \vec{r}_{v E} \\ \dot{\vec{r}}_{s_E} &= \dot{\vec{r}}_{v s_E} + \dot{\vec{r}}_{v av} \end{aligned} \quad (18)$$

- 7) It is necessary to evaluate the conic section time of flight so that  $\kappa$  may be found to use in the next iteration. First, some preliminary orbit variables must be determined. The in-plane normal to the major axis is

$$\vec{n} = \begin{cases} \frac{\vec{k} \times \vec{e}}{k e}, & e \neq 0 \\ \frac{\vec{k} \times \vec{r}_{v s_B}}{k r_{v s_B}}, & e = 0 \end{cases} \quad (19)$$

The length of the semimajor axis is given by

$$\begin{aligned} b &= \frac{k^2}{\mu_{v av} (|1 - e^2|)^{1/2}}, \quad e \neq 1 \\ b_i &= \frac{2}{r_{v s_i} - k^2 / \mu_{v av}}, \quad e = 1, \quad i = B, E \end{aligned} \quad (20)$$

The projection of the radius vector orthogonal to the major axis, divided by  $b$  is given by

$$X_i = \frac{\vec{n} \cdot \vec{r}_{v s_i}}{b_i}, \quad i = B, E \quad (21)$$

The mean angular rate is

$$\omega_M = \begin{cases} \frac{\mu_{v av} (1 - e^2)}{k b}, & e \neq 1 \\ \frac{k}{2}, & e = 1 \end{cases} \quad (22)$$

where  $\omega_M < 0$  for hyperbolic orbits. The eccentric anomaly is given by,

$$E_i = \begin{cases} \sin^{-1} X_i, & e < 1 \\ \frac{k^2 / \mu_{v av} \cdot X_i}{3}, & e = 1 \\ \sinh^{-1} X_i, & e > 1 \end{cases}, \quad i = B, E \quad (23)$$

Then

$$M_i = E_i - e X_i, \quad i = B, E \quad (24)$$

and finally the conic section time of flight is given as

$$\Delta\tau = t_2 - t_1 = \frac{M_E - M_B}{\omega_M} \quad (25)$$

The intermediate variable  $\kappa$ , to be used in the next interval is calculated as

$$\kappa = \frac{\Delta\tau - \Delta t}{(\Delta t)^2} \quad (26)$$

- 8) The final positions and velocities of the planets are now calculated from the ephemeris subroutine and returned to the virtual mass routine.
- 9) The final position and mass magnitude of the virtual mass are now recalculated from the assumed position of the spacecraft at the end of the interval and the planetary ephemerides by using equation (2). The velocity and magnitude rate of the virtual mass are computed from

$$\begin{aligned}\dot{\vec{r}}_{vE} &= \frac{\dot{\vec{M}} - \vec{r}_{vSE} \dot{M}_S}{M_S} \\ \dot{\mu}_v &= \mu_{vE} \left[ \frac{V_{vSE}}{r_{vSE}} + \frac{\dot{M}_S}{M_S} \right]\end{aligned}\tag{27}$$

where

$$\begin{aligned}\dot{\vec{M}} &= \sum_{i=1}^n \frac{\mu_i}{r_{iSE}} \left[ \dot{\vec{r}}_{iE} - \vec{r}_{iE} \left( \frac{V_{iSE}}{r_{iSE}} \right) \right] \\ M_S &= \sum_{i=1}^n \frac{\mu_i}{(r_{iSE})^3} \left( \frac{V_{iSE}}{r_{iSE}} \right) \\ \frac{V_{iSE}}{r_{iSE}} &= \frac{3 \vec{r}_{iSE} \cdot \dot{\vec{r}}_{iSE}}{(r_{iSE})^2}\end{aligned}$$

- 10) After this last computation, Step 9, one complete iteration has been obtained. The values of the final position and magnitude of the virtual mass that was just calculated is compared to the one assumed in the computation of the spacecraft trajectory. If they do not

agree to within a set tolerance, the new values of  $\vec{r}_{vE}$ ,  $\mu_{vE}$ , and  $k$  are returned to Step 3 and another iteration is performed. However, it should be pointed out that Step 8 is to be omitted from future iterations. The final positions of the planets do not differ from one iteration to the next since the final time is fixed.

- 11) After two iterations, the required quantities are stored for the next time interval and the algorithm returns to Step 1.

More complete details for the above computational algorithm may be found in Novak's report (ref. 1). One point worth mentioning at this juncture concerns the use of the words "accuracy level" when referring to an orbit computed by the trajectory mode. As was mentioned earlier, the step size used in the virtual mass calculation of n-body trajectories refers to the true anomaly arc, with respect to the virtual mass, that is kept fixed throughout the trajectory. Thus, a fixed true anomaly arc of 1 mrad means that each individual computing interval, using the algorithm defined above, results in a two-body arc around the effective force center of 1 mrad. If the fixed true anomaly arc is 10 mrad, then clearly fewer computational intervals are used and the resulting trajectory, neglecting computer noise, is less accurate.

An external accuracy level is input to the program, where this value is subsequently changed into a fixed true anomaly arc for the computing intervals. An external accuracy of  $2.5 \times 10^{-5}$ , for example, corresponds to a true anomaly arc of 16.57 mrad; similarly, an external accuracy level of  $1 \times 10^{-6}$  corresponds to a fixed true anomaly of 3.6 mrad. For the n-body problem the external accuracy level is a dummy variable; it was initially set up to represent the accumulated percentage position error for a restricted three-body problem after one orbit. Thus lower accuracy levels imply lower amounts of arc for fixed time anomaly used in the computations and, consequently, more accurate trajectories that require more computer time. Throughout this report, when referring to the computational interval size used by the trajectory mode or subroutine, the phase accuracy level is employed.

#### IV. TARGETING MODE - ANALYSIS

##### A. Statement of Problem

The mathematical formulation of the targeting problem results in a classical two point boundary value problem. The motion of the infinitesimal probe moving under the gravitational attractions of n-bodies is governed by the nonlinear differential system

$$\begin{aligned}\dot{\vec{r}} &= \vec{V} \\ \dot{\vec{V}} &= \sum_{i=1}^n \mu_i \left( \frac{\vec{r}_i(t) - \vec{r}}{|\vec{r}_i(t) - \vec{r}|^3} \right)\end{aligned}\quad (28)$$

where  $\mu_i$  and  $\vec{r}_i(t)$  are respectively the (known) gravitational constant and position vector of the  $i^{\text{th}}$  gravitational body. An initial time  $t_0$  and a target time  $t_T$  are specified along with a set of target conditions

$$\vec{r}(t_T), \vec{V}(t_T), t_T = 0 \quad (29)$$

The solution of the targeting problem is the determination of a set of initial (injection) conditions  $\vec{r}_0, \vec{V}_0$  such that integration of the system (28) using the initial conditions

$$\begin{aligned}\vec{r}(t_0) &= \vec{r}_0 \\ \vec{V}(t_0) &= \vec{V}_0\end{aligned}\quad (30)$$

yields solution values at  $t = t_T$  consistent with the target constraints (29).

The system (28) being of sixth order implies that (generally) six independent constraints define a solution. The target constraints to be considered in this analysis are either two or three-dimensional. Hence to ensure that the problems are not underdetermined, additional constraints must be placed on the initial conditions (30). Therefore the injection position  $\vec{r}_0$  is forced to satisfy constraints consistent with the general mission requirements and a standard launch profile originating from Cape Kennedy. In the case of a two dimensional target constraint similar restrictions are placed on the z-component of injection velocity. With these additions the targeting problem is well-defined.

The standard procedure may now be summarized. From the general mission requirements (launch data, encounter planet, encounter time) a realistic injection position vector (and z-component of injection velocity) is computed. Using these as constraints along with the exact target conditions (29), the injection velocity  $\vec{v}_0$  is determined which solves the two point boundary value problem (28), (29), and (30).

#### B. General Method of Solution

A comprehensive survey of current techniques applicable to the two-point boundary value problem was made before selecting the numerical differencing algorithm as the approach to be used in the targeting program. The alternative techniques considered will be presented before proceeding to a discussion of the algorithm chosen.

Five general approaches were considered applicable to the solution of the targeting problem but inferior to the numerical differencing scheme ultimately selected. These candidate methods included the adjoint method, the perturbation function method, quasi-linearization, invariant imbedding, and patched conic approximations.

In the adjoint (ref. 2) and perturbation function (ref. 3) methods, linearized equations (specifically the adjoint and perturbation equations) governing the dynamics of small perturbations about a current iterate would be integrated to determine corrections to that iterate. However the inclusion of linear differential equations would necessitate an additional integration program as the virtual mass routine could not perform this integration. The time constraints in the target conditions also add a great deal of complexity when using these methods. Furthermore these techniques offer no apparent advantages over numerical differencing.

In quasi-linearization (ref. 4), the approach is to replace the nonlinear system of equations by an approximating linear system (linearized about the current iterate) so that the relative simplicity of this revised system can be exploited to yield improved injection conditions. Quasi-linearization has better convergence properties than any of the other proposed methods but unfortunately suffers from serious defects in this application. The state of the current iterate at any time along the trajectory must be available to the integrator. Since an interplanetary trajectory requires not only a large number of integration steps, but also a varying size for these steps, this represents a burden both in computer storage and computation. In addition, quasi-linearization requires an independent integrator.

In invariant imbedding (ref. 5), the two-point boundary value problem is replaced by a system of first order partial differential equations that are solved successively in one-directional sweeps. Its main advantage lies in the fact that it is not an iterative procedure. However the application of this method to such a multidimensional problem introduces great analytical and numerical difficulties. Furthermore there is no control on the final tolerances of target errors.

The last method tried involved using patched conic approximations to estimate the velocity corrections iteratively. Thus the method would employ analytic tools such as Danby's matrizant (ref. 6). This would be an acceptable approach if the target conditions were all position or velocity dependent. However, because time appears explicitly in the target constraints and because there are no formal analytical time relationships, this approach was also discarded.

The numerical differencing algorithm is conceptually quite simple. It is an iterative scheme that generates progressive sets of injection velocities, each set leading to reduced errors in the target conditions. The procedure therefore requires an initial approximation to the injection velocity. This is no problem, however, as the technique used to compute the injection position vector also generates an acceptable zero iterate injection velocity.

Let the target conditions be denoted  $\vec{\tau} = (\tau_1, \tau_2, \tau_3)$ . Let  $\vec{r}_0$  be the injection position vector and let

$$\vec{V}(k) = \left( V_x(k), V_y(k), V_z(k) \right)$$



be the  $k^{\text{th}}$  iterate of the injection velocity. Let the integration of the equations of motion (28) using  $(\vec{r}_0, \vec{v}^{(k)})$  as injection conditions yield target values  $\vec{T} = (T_1, T_2, T_3)$ . If the error  $\Delta\vec{T} = \vec{T} - \vec{T}$  is acceptable, then  $\vec{v}^{(k)}$  is returned as the desired solution.

If the error  $\Delta\vec{T}$  is unacceptable, the numerical differencing algorithm generates a correction  $\Delta\vec{V} = (\Delta V_x, \Delta V_y, \Delta V_z)$  to  $\vec{v}^{(k)}$ , which should lead to improved target conditions. The trajectory target values are obviously functions of the injection velocity  $\vec{V}$  defining the trajectory. Thus

$$\vec{T} = \vec{T}(\vec{V})$$

Expanding this in a Taylor series about the  $k^{\text{th}}$  iterate injection velocity and retaining only linear terms yields the approximation

$$\vec{T}[\vec{v}^{(k)} + \Delta\vec{V}] = \vec{T}[\vec{v}^{(k)}] + \frac{\partial \vec{T}}{\partial \vec{V}}[\vec{v}^{(k)}] \cdot \Delta\vec{V} \quad (31)$$

where

$$\frac{\partial \vec{T}}{\partial \vec{V}}[\vec{v}^{(k)}] = \begin{bmatrix} \frac{\partial T_1}{\partial V_x}[\vec{v}^{(k)}] & \frac{\partial T_1}{\partial V_y}[\vec{v}^{(k)}] & \frac{\partial T_1}{\partial V_z}[\vec{v}^{(k)}] \\ \frac{\partial T_2}{\partial V_x}[\vec{v}^{(k)}] & \frac{\partial T_2}{\partial V_y}[\vec{v}^{(k)}] & \frac{\partial T_2}{\partial V_z}[\vec{v}^{(k)}] \\ \frac{\partial T_3}{\partial V_x}[\vec{v}^{(k)}] & \frac{\partial T_3}{\partial V_y}[\vec{v}^{(k)}] & \frac{\partial T_3}{\partial V_z}[\vec{v}^{(k)}] \end{bmatrix}$$

Implicit in this equation is the assumption that the required velocity change lies in a linear region in the injection condition - target condition manifold.

The partials  $\frac{\partial T_i}{\partial V_\alpha}$  are now approximated by numerical differencing (from which the procedure derives its name). The equations of motion are integrated using as initial conditions

$$\begin{aligned}\vec{r}(t_0) &= \vec{r}_0 \\ V_x(t_0) &= V_x^{(k)} + \delta V_x \\ V_y(t_0) &= V_y^{(k)} \\ V_z(t_0) &= V_z^{(k)}\end{aligned}$$

Because of the (small) perturbation in  $V_x(t_0)$ , the resulting target values will differ slightly from  $\vec{T}^{(k)}$ ; denote them by  $\vec{T}^{(k)} + \delta\vec{T} = (T_1 + \delta T_1, T_2 + \delta T_2, T_3 + \delta T_3)$ . Then by estimating the differentials by the numerical differences, the approximations result

$$\frac{\partial T_1}{\partial V_x} = \frac{\delta T_1}{\delta V_x}, \quad \frac{\partial T_2}{\partial V_x} = \frac{\delta T_2}{\delta V_x}, \quad \frac{\partial T_3}{\partial V_x} = \frac{\delta T_3}{\delta V_x}$$

A similar procedure is used to evaluate the remaining partial derivatives occurring in equation (31). Denoting

$$\Phi = \begin{bmatrix} \frac{\delta T_1}{\delta V_x} & \frac{\delta T_1}{\delta V_y} & \frac{\delta T_1}{\delta V_z} \\ \frac{\delta T_2}{\delta V_x} & \frac{\delta T_2}{\delta V_y} & \frac{\delta T_2}{\delta V_z} \\ \frac{\delta T_3}{\delta V_x} & \frac{\delta T_3}{\delta V_y} & \frac{\delta T_3}{\delta V_z} \end{bmatrix}$$

Equation (31) may now be rewritten

$$\vec{T} \left[ \vec{V}^{(k)} + \Delta \vec{V} \right] = \vec{T} \left[ \vec{V}^{(k)} \right] + \Phi \Delta \vec{V}$$

The correction  $\Delta \vec{V}$  is to be chosen so that  $\vec{T} \left[ \vec{V}^{(k)} + \Delta \vec{V} \right] = \vec{\tau}$ . Substituting this into the above equation and rearranging yields

$$\Delta \vec{\tau} = \vec{\tau} - \vec{T} = \Phi \Delta \vec{V}$$

If  $\Phi$  is nonsingular the preceding equation may be solved for the correction  $\Delta \vec{V}$  resulting in

$$\Delta \vec{V} = \Phi^{-1} \Delta \vec{\tau}$$

and the  $(k + 1)$  st iterate is then given by  $\vec{V}^{(k + 1)} = \vec{V}^{(k)} + \Delta \vec{V}$ . The entire process is repeated until an acceptable iterate is generated.

It is important to recall the assumptions made in this model to pinpoint where trouble might arise and cause the procedure to fail (or diverge). The linearity assumption in deriving equation (31) is probably the major problem. If that assumption is bad, the entire analysis is in error. If the magnitude of the required  $\Delta \vec{V}$  is large, the omission of the second order terms in  $\Delta \vec{V}$  in equation (31) would again be unjustified. However experience has shown this to be an unusual event. The approximation of the partials by their corresponding numerical differences is another potential problem. However the fact that the program allows the increment  $\delta V$  by which the velocities are perturbed to be input permits the selection of a reasonable  $\delta V$  for this computation. Finally, the matrix  $\Phi$  may be singular. Again, experience has shown this to be quite rare. If this did occur changing the magnitude of the  $\delta V$  or reading in perturbed values for the injection velocity could remove the problem.

One safeguard is included in the program to minimize the effects of any of the above faults. A scalar measure is assigned to the target conditions of each nominal. If the error in the resulting correction is increased, the correction is reduced by a quarter. The process is repeated until a correction is found that does decrease the error.

### C. General Definition of Target Conditions

The four basic target conditions are divided into two classes: the sphere-of-influence conditions and the closest-approach conditions. The sphere-of-influence conditions are based on the impact plane parameters  $\vec{B} \cdot \vec{T}$  and  $\vec{B} \cdot \vec{R}$  introduced by Kizner (ref. 7). Because there is some ambiguity in the general definitions of these parameters, the exact parameters as used throughout this report will now be defined (fig. 1).

From the point of intersection of the probe trajectory with the target planet sphere-of-influence the trajectory is assumed to be hyperbolic with the target planet as the prime body. One of three primary axes, the  $\hat{S}$ -axis, is constructed parallel to the incoming asymptote and passing through the center of the target planet. The impact plane is constructed perpendicular to the  $\hat{S}$ -axis and passing through the center of the target planet. The intersection of the ecliptic plane with the impact plane forms

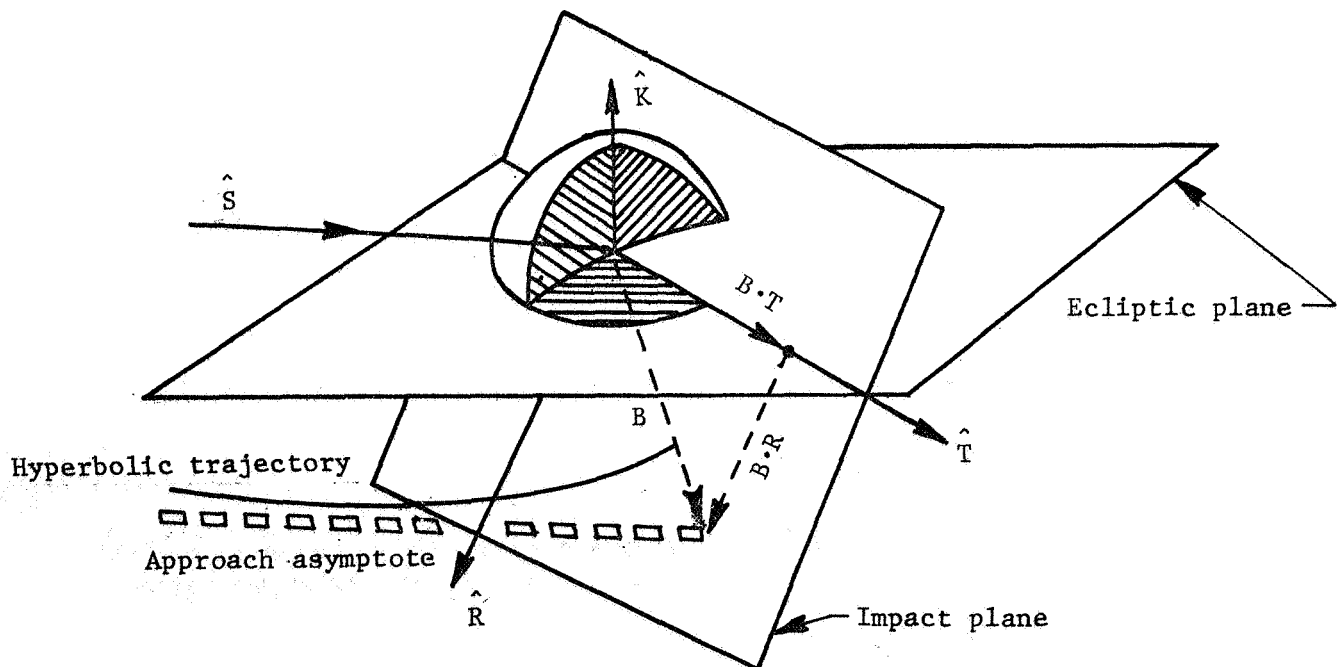


Figure 1. Impact Plane Parameters

a line that defines the  $\hat{T}$ -axis. For definiteness, if  $\hat{K}$  is the planet normal to the ecliptic plane, then the direction of  $\hat{T}$  is governed by  $\hat{S} \times \hat{K}$ . Finally, the  $\hat{R}$ -axis is defined to make the RST system right-handed, i.e.,  $\hat{R} = \hat{S} \times \hat{T}$ . The miss parameter  $\vec{B}$  is the vector lying in the impact plane from the origin to the intersection of the approach asymptote with that plane. The impact plane parameters  $\vec{B} \cdot \hat{T}$  and  $\vec{B} \cdot \hat{R}$  are now given by the standard definitions. The third sphere-of-influence condition is the time at which the sphere-of-influence is encountered by the probe. In the first major targeting option, the parameters  $\vec{B} \cdot \hat{T}$  and  $\vec{B} \cdot \hat{R}$  are used as targets while  $t_{SI}$  is met only approximately. Thus  $\tau_1 = \vec{B} \cdot \hat{R}$  and  $\tau_2 = \vec{B} \cdot \hat{T}$  and the matrix  $\Phi$  appearing in equation (31) is only  $2 \times 2$ . In the second option all three constraints are met, i.e.,  $\tau_1 = \vec{B} \cdot \hat{T}$ ,  $\tau_2 = \vec{B} \cdot \hat{R}$ , and  $\tau_3 = t_{SI}$ , and  $\Phi$  is  $3 \times 3$ .

The definitions of the closest-approach conditions are much more straightforward. The first parameter  $r_{CA}$  is the radius of closest approach. This is simply the minimal distance between the probe and the target planet during the trajectory. The vector position and velocity at this point (with respect to the target planet) are used to compute the instantaneous orbital plane. The angle between this orbital plane and the planet's equatorial plane defines the second parameter  $i_{CA}$ , the inclination at closest approach. Finally the last parameter is  $t_{CA}$ , the time at closest approach. In the third of the four basic options the three target conditions  $\tau_1 = r_{CA}$ ,  $\tau_2 = i_{CA}$ ,  $\tau_3 = t_{CA}$  are met approximately; in the fourth they are met to specified tolerances determined by the input.

#### D. Generation of Patched Conic Injection Conditions

The injection position vector and the zero-iterate injection velocity vector are computed from a patched conic approximation to the interplanetary trajectory using a realistic launch profile from Cape Kennedy. These injection conditions are not patched conic conditions in the strict sense; when propagated forward in a patched conic they do not satisfy the mission requirements. A bias is included in the injection velocity which, while corrupting the patched conic model, more nearly approximates the velocity required by an

n-body trajectory satisfying the mission. To distinguish these conditions from actual patched conic conditions, they are termed point-to-point conditions. The point-to-point conditions are generated in the SPARC program developed by JPL. A rigorous analysis may be found in reference 8. However, for completeness, the relevant details will be repeated here.

1. Heliocentric phase.— Given a launch date  $t_1$ , a launch planet  $P_1$ , an encounter date  $t_2$ , and an encounter planet  $P_2$ , a rough approximation to the interplanetary trajectory is easily obtained. The assumption is first made that the trajectory is governed only by the gravitational attraction of the sun, and therefore may be represented by a conic having the sun at one focus. Let the position of  $P_1$  with respect to the sun at the time  $t_1$  be denoted  $\vec{R}_1$ . Similarly define  $\vec{R}_2$  as the position of  $P_2$  at the time  $t_2$ .

The heliocentric central angle  $\psi$  is defined by

$$\cos \psi = \vec{R}_1 \cdot \vec{R}_2 (R_1 R_2)^{-1}$$

$$\sin \psi = \text{sgn} \left| (\vec{R}_1 \times \vec{R}_2) \cdot \hat{K} \right| (1 - \cos^2 \psi)^{\frac{1}{2}}$$

where  $\hat{K}$  is the unit normal to the heliocentric ecliptic plane. The unit vector  $\hat{W}$  normal to the probe's orbital plane is now defined by

$$\hat{W} = \vec{R}_1 \times \vec{R}_2 (R_1 R_2 \sin \psi)^{-1}$$

Finally, the inclination of the orbital plane with respect to the ecliptic plane is given by

$$i = \cos^{-1} (\hat{W} \cdot \hat{K}) \quad 0 \leq i \leq \pi$$

The semimajor axis  $a$  of the heliocentric ellipse is determined from Lambert's theorem: the transfer time between any two points on an ellipse is a function of the sum of the distances of each point from the focus, the distance between the points, and the semimajor axis of the ellipse. Thus

$$t_2 - t_1 = f(R_1 + R_2, |\vec{R}_1 - \vec{R}_2|, a)$$

This equation is solved iteratively to obtain the semimajor axis  $a$ . Battin (ref. 9) has demonstrated that the eccentricity  $e$  is actually a function of  $a$ . Using his procedure the eccentricity is computed.

The state of the probe at any point along the heliocentric trajectory may now be computed. The true anomaly  $v$  corresponding to a given radius  $R$  is given by

$$\cos v = \frac{a(1-e^2) - R}{eR}$$

The path angle  $\Gamma$  at that point is then determined from

$$\sin \Gamma = R^{\frac{1}{2}} (1-e^2)^{-\frac{1}{2}} (2a-R)^{-\frac{1}{2}} e \sin v$$

The heliocentric speed is given by  $V = \left[ \mu_s \left( 2R^{-1} - a^{-1} \right) \right]^{\frac{1}{2}}$  where  $\mu_s$  is the gravitational constant of the sun. The vector velocity is then given by

$$\vec{V} = VR^{-1} \left[ \left( \hat{W} \times \vec{R} \right) \cos \Gamma + \vec{R} \sin \Gamma \right]$$

This allows the computation of the velocity  $\vec{V}_1$  of the probe (with respect to the sun) along its trajectory at the point of departure from the launch planet. The hyperbolic excess velocity  $\vec{V}_{hL}$  is then determined from

$$\vec{V}_{hL} = \vec{V}_1 - \vec{V}_{LP}$$

where  $\vec{V}_{LP}$  is the heliocentric velocity of the launch planet at the time of launch.

2. Near launch planet phase.— Having computed the approximate hyperbolic excess velocity required for the mission, the near launch planet trajectory may be computed.

The unit vector in the direction of the outgoing asymptote is

$$\hat{S} = V_{hL}^{-1} \vec{V}_{hL}$$

The unit vector  $\hat{N}$  normal to the trajectory plane is then constrained by the requirements  $\hat{N} \cdot \hat{N} = 1$  and  $\hat{N} \cdot \hat{S} = 0$ . The components of  $\hat{N}$  are thus restricted by

$$N_z = \cos \phi_L \sin \Sigma_L$$

$$N_y = \left\{ -N_z S_y S_z \pm S_x \left[ 1 - (S_z^2 + N_z^2) \right]^{\frac{1}{2}} \right\} (S_x^2 + S_y^2)$$

$$N_x = - (N_y S_y - N_z S_z) S_x^{-1}$$

where  $\phi_L$  is the latitude of the launch site and  $\Sigma_L$  is the launch azimuth. The second of these equations restricts the launch azimuth to satisfy

$$\sin^2 \Sigma_L \leq (1 - S_z^2) \cos^{-2} \phi_L$$

In this program the "short coast" option is selected by choosing the ambiguous sign in the second component equation to be negative. A "due east" launch is generated when possible by setting  $\Sigma_L = 90^\circ$ .

If this violates the above constraint, the nearest possible azimuth is chosen.

The eccentricity of the orbit is determined from the equation

$$e = 1 + R_p V_{hL}^2 \mu^{-1}$$

where  $R_p$ , the periapsis radius, is set internally and will be discussed in a subsequent section. The true anomaly of the outgoing asymptote is given by

$$\cos v_s = -e^{-1}$$

In essence this defines the position of the periapsis. The standard unit vectors  $\hat{P}$  (toward periapsis) and  $\hat{Q}$  (normal to  $\hat{P}$  in the plane of the trajectory) are defined by



$$\hat{P} = \hat{S} \cos v_s + (\hat{S} \times \hat{N}) \sin v_s$$

$$\hat{Q} = \hat{S} \sin v_s - (\hat{S} \times \hat{N}) \cos v_s$$

These vectors now determine the near launch trajectory. For a given true anomaly  $v$  the magnitudes of the vector position and velocity are respectively

$$R = p \cdot (1 + e \cos v)^{-1}$$

$$V = (C_3 + 2\mu_p R^{-1})^{\frac{1}{2}}$$

where  $C_3 = \frac{1}{2} V_{hL}^2$ . The normalized position and velocity vectors are

$$\hat{R} = \hat{P} \cos v + \hat{Q} \sin v$$

$$\hat{V} = -\hat{P} \sin (v - \Gamma) + \hat{Q} \cos (v - \Gamma)$$

where the path angle  $\Gamma$  is defined by

$$\cos \Gamma = (p\mu_p) V^{-1} R^{-1}, \quad -\frac{\pi}{2} \leq \Gamma \leq \frac{\pi}{2}.$$

The true anomaly at injection  $v_I$  is an internally supplied constant. Hence the injection position and velocity may be easily computed using the equations of the previous section.

The angle  $\psi_{LSP}$  between the launch site and the orbit periaapsis is given by

$$\cos \psi_{LSP} = \hat{R}_{LS} \cdot \hat{P}$$

$$\sin \psi_{LSP} = \hat{R}_{LS} \cdot \hat{Q} \quad 0 \leq \psi_{LSP} \leq 2\pi$$

where  $\hat{R}_{LS}$  is the unit vector to the launch site =  $(\cos\phi_L \cos\theta_L, \cos\phi_L \sin\theta_L, \sin\phi_L)$ . The angle  $\psi_I$  between launch and injection is then  $\psi_I = 2\pi - \psi_{LSP} - \psi_I$ . The coast time is then calculated from  $T_c = \left| \psi_I - (\psi_1 + \psi_2) \right| k_\phi$ . Here  $\psi_1$  and  $\psi_2$  are the angles of first and second burn respectively, and  $k_\phi$  is the inverse of the parking orbit coast rate and all are set internally in the program.

The injection latitude  $\phi_I$  is given by  $\sin \phi_I = (\hat{R}_I)_z$ ,  $-\frac{\pi}{2} \leq \phi_I \leq \frac{\pi}{2}$ . The right ascension of injection  $\theta_I$  is defined by

$$\cos \theta_I = (R_I)_x \left[ (R_J)_x^2 + (R_I)_y^2 \right]^{-\frac{1}{2}}$$

$$\sin \theta_I = (R_I)_y \left[ (R_J)_x^2 + (R_I)_y^2 \right]^{-\frac{1}{2}}$$

The injection longitude is given by

$$\theta_I = \theta_I - \theta_L - \omega t_b + \theta_L$$

where  $\theta_L$  is the longitude of the launch site and  $\omega t_b$  is the time from launch to injection =  $t_1 + T_c + t_2$ . The injection azimuth angle is calculated from

$$\cos \Sigma_I = \frac{S_z - \cos'(\psi_s - \psi_I) \sin \phi_I}{\sin(\psi_s - \psi_I) \cos \phi_I}$$

The launch time is given by

$$T_L = \omega^{-1} (\theta_L - \theta_L - GHA) \bmod 2\pi$$

where GHA is the Greenwich hour angle at  $0^h$  UT of the launch day. Specifically,  $GHA = 100.07554260^\circ + 0.985647346^\circ T_d +$

$2.9015 \times 10^{-13} T_d^2$ ,  $0 \leq GHA \leq 2\pi$ , where  $T_d$  = days past  $0^h$  January 1, 1950. The injection time is calculated from

$$T_I = T_L + t_b$$

3. Internally supplied constants.— Several of the parameters discussed previously are set internally in the program. These parameters and their values are listed in the following tabulation.

<u>Symbol</u>	<u>Definition</u>	<u>Target = Inner Planet</u>	<u>Target = Outer Planet</u>
$\psi_1$	Angle of first burn	$17^\circ$	$23^\circ$
$t_1$	Time of first burn	500 sec	700 sec
$\psi_2$	Angle of second burn	$8^\circ$	$25^\circ$
$t_2$	Time of second burn	100	300
$k_\phi$	Inverse parking orbital rate	14.689	14.689
$v_I$	True anomaly in hyperbolic orbit at injection	$3.7^\circ$	$12.0^\circ$
$R_p$	Perifocal distance of escape hyperbole = parking orbit radius	6560 km	6560 km
$\phi_L$	Latitude of launch site	$28.317^\circ$	$28.28^\circ$
$\theta_L$	Longitude of launch site	$279.457^\circ$	$279.5^\circ$

The values used here are chosen so that the point-to-point conditions agree with the published SPARC data on Venus and Mars missions. However, they may be easily changed to any values desired.

## E. Auxiliary Computations in Targeting

1. Computation of impact plane parameters.— The impact plane parameters  $\vec{B} \cdot \hat{T}$  and  $\vec{B} \cdot \hat{R}$  are functions of the position and velocity vectors  $\vec{r}_{SI}$  and  $\vec{v}_{SI}$  (with respect to the target planet) recorded at the point of intersection of the interplanetary trajectory with the target planet sphere of influence. The computation proceeds as follows:

$$c = |\vec{r}_{SI} \times \vec{v}_{SI}|$$

$$\dot{\rho} = r_{SI}^{-1} \vec{r}_{SI} \cdot \dot{\vec{v}}_{SI}$$

$$p = c^2 \mu_P^{-1}$$

$$a = \mu_P r_{SI} (2\mu - r_{SI} v_{SI}^2)^{-1}$$

$$e = 1 - pa^{-1}$$

$$\cos v_{SI} = e^{-1} r_{SI}^{-1} (P - r_{SI})$$

$$\sin v_{SI} = \mu_P^{-1} e^{-1} \dot{\rho} c$$

$$b = p^{\frac{1}{2}} |a|^{\frac{1}{2}}$$

$$\vec{Z} = c^{-1} \left| r_{SI} \vec{v}_{SI} - \dot{\rho} \vec{r}_{SI} \right|$$

$$\hat{P} = r_{SI}^{-1} \cos v_{SI} \vec{r}_{SI} - \sin v_{SI} \vec{Z}$$

$$\hat{Q} = r_{SI}^{-1} \sin v_{SI} \vec{r}_{SI} + \cos v_{SI} \vec{Z}$$

$$\hat{S} = (a^2 + b^2)^{-1} \left| -a \hat{P} + b \hat{Q} \right|$$

$$\hat{B} = (a^2 + b^2)^{-1} \begin{vmatrix} b^2 \hat{P} + ab\hat{Q} \end{vmatrix}$$

$$T_1 = S_2 (a^2 + b^2)^{-\frac{1}{2}}$$

$$T_2 = -S_1 (a^2 + b^2)^{-\frac{1}{2}}$$

$$T_3 = 0$$

$$\hat{R} = \hat{S} \times \hat{T}$$

Now  $\vec{B} \cdot \hat{T}$  and  $\vec{B} \cdot \hat{R}$  are defined by the standard formulas.

2. Computation of closest approach conditions.— In computing  $r_{CA}$ ,  $i_{CA}$ , and  $t_{CA}$  to construct the closest approach state transition matrix, it is necessary to compute the exact time of closest approach. Otherwise, the state transition matrix may become invalid. The virtual mass integration program records the values  $r_{CA}$ ,  $V_{CA}$ ,  $t_{CA}$  when the distance from the target planet to the probe first begins to increase. Since a more accurate evaluation is required, the following iterative scheme is used to refine these estimates: Now  $\dot{r} = \vec{r} \cdot \vec{V}$ . Differentiating yields  $\ddot{r} + r\ddot{r} = V^2 + \vec{r} \cdot \ddot{\vec{V}}$ . Now replacing  $\ddot{\vec{V}}$  by  $-\mu\vec{r}r^{-3}$  leads to

$$\ddot{r} = r^{-1} \left[ V^2 - \mu r^{-1} - \dot{r}^2 \right]$$

Approximating  $\dot{r}(t + \Delta t) = \dot{r}(t) + \ddot{r}(t) \Delta t$ , the correction  $\Delta t$  which should result in  $\dot{r}(t_C + \Delta t) \leq \dot{r}(t_C)$  is given by

$$\Delta t = \frac{-\dot{r}(t_C)}{\ddot{r}(t_C)}$$

where  $\ddot{r}(t_C)$  is given above. Although obtaining a better estimate of  $t_C$  is most important, the correction to  $\vec{r}_{CA}$  is also made:

$$\left( \vec{r}_{CA} \right)_{\text{new}} = \vec{r}_{CA} + \vec{V}_{CA} \cdot \Delta t$$

### 3. Conversion of closest approach to sphere of influence targets.-

In both the third and fourth targeting options  $(r_{CA}, i_{CA}, t_{CA})$  it is efficacious to first translate the target conditions to equivalent conditions in  $\vec{B} \cdot \hat{T}$ ,  $\vec{B} \cdot \hat{R}$ ,  $t_{SI}$  and target first to these auxiliary conditions. The computations for this conversion are based on patched conic trajectories. Figure 2 illustrates the problem.

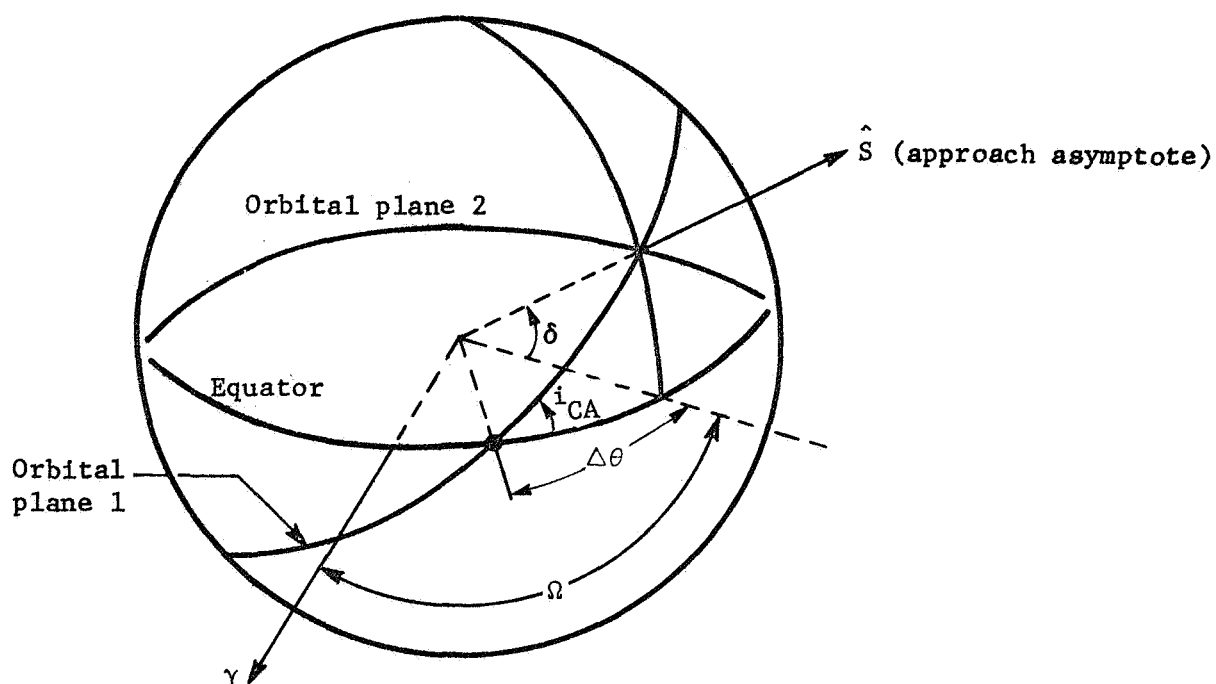


Figure 2.- Spherical Trigonometry for Closest Approach Conversion

Let the planeocentric ecliptic position and velocity vectors at arrival at the target planet sphere of influence be given by  $\vec{r}_{SI}$  and  $\vec{v}_{SI}$ . Let the target conditions be denoted  $r_{CA}$ ,  $i_{CA}$ ,  $t_{CA}$ . Let the matrix defining the transformation from equatorial

coordinates to ecliptic coordinates be given by  $M_{EQEC}$ . The approach asymptote may be approximated quite well by  $\vec{V}_{SI}$ . Converting to equatorial coordinates this becomes

$$\hat{S} = M_{EQEC} \frac{\vec{V}_{SI}}{V_{SI}}$$

The (equatorial) declination and right ascension of  $S$  are given by

$$\delta = \tan^{-1} S_3$$

$$\theta = \tan^{-1} (S_2/S_1)$$

Figure 2 illustrates that for a given inclination  $i_{CA}$ ,  $0 \leq i_{CA} \leq \frac{\pi}{2}$ , there are four possible orbits having that inclination and including  $\hat{S}$  in the plane of motion. The trajectories could lie in either of orbital planes 1 or 2 and be either posigrade or retrograde. To avoid ambiguity, the program considers  $i = i_{CA}$  (under the constraint  $0 \leq i_{CA} \leq \pi/2$ ) to require the posigrade orbit in orbital plane 1 detailed in the figure;  $i = -i_{CA}$  specifies the posigrade orbit in orbital plane 2, and  $i = \pm (\pi - i_{CA})$  prescribes the two retrograde orbits. Interest will be focused only on the illustrated case for the purpose of discussion.

Consideration of Figure 2 demonstrates the target inclination  $i_{CA}$  must be less than the declination of the incoming asymptote, i.e.,

$$i_{CA} \leq \delta$$

If this constraint is not satisfied, the target inclination is set equal to the declination before proceeding. The angle  $\Delta\theta$  is deduced from spherical trigonometry as

$$\Delta\theta = \sin^{-1} \left( \frac{\tan \delta}{\tan t_{CA}} \right)$$

Therefore, the ascending node of the orbit (with respect to the equator) is given by

$$\Omega = \Theta - \Delta\theta$$

The normal  $\hat{C}$  to the orbital plane in equatorial coordinates may now be written

$$C_1 = S_3 \sin \Omega \quad C_2 = -S_3 \cos \Omega \quad C_3 = S_2 \cos \Omega - S_1 \sin \Omega$$

The matrix  $M_{EQIM}$  defining transformations from the equatorial plane to the impact plane is then constructed. The normal to the orbital plane in impact plane coordinates  $\hat{C}_{IMP} = M_{EQIM} \hat{C}$  may then be used to compute  $\beta$ , the angle in the B-plane from the T-axis to the impact parameter  $\vec{B}$  (see figure 3).  $\beta$  is then given by

$$\beta = \frac{\pi}{2} + \tan^{-1} \left[ \frac{(C_{IMP})_2}{(C_{IMP})_1} \right]$$

The magnitude of  $\vec{B}$  is given quite simply as

$$B = r_{CA} \left( 1 + 2\mu_p r_{CA}^{-1} v_{SI}^{-2} \right)$$

Then the target  $B \cdot T$  and  $B \cdot R$  are given by

$$\vec{B} \cdot \hat{T} = B \cos \beta$$

$$\vec{B} \cdot \hat{R} = B \sin \beta$$



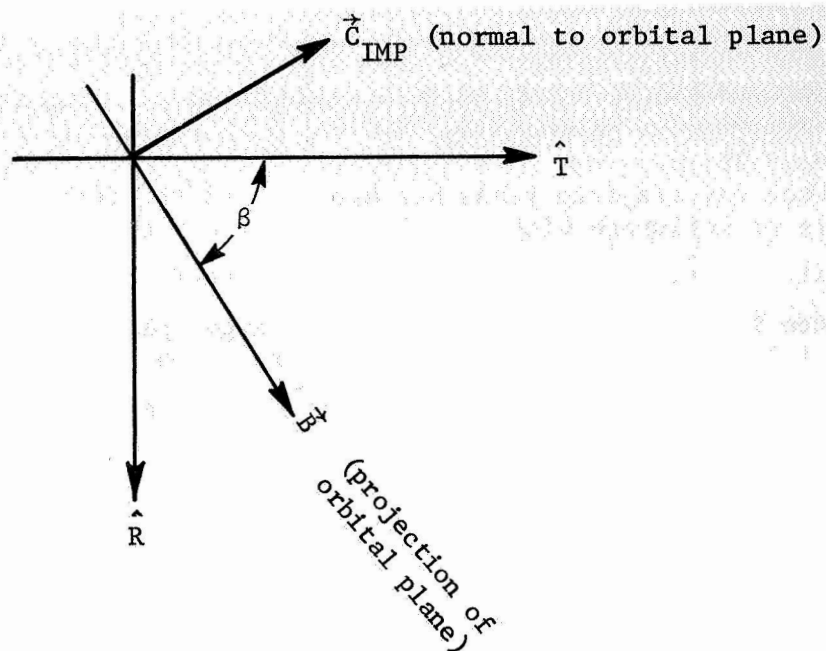


Figure 3.- Projection of Orbital Plane and Normal on Impact Plane

To translate the time at closest approach  $t_{CA}$  into a consistent time at sphere of influence  $t_{SI}$  the elements of the hyperbola determined by the state  $\vec{r}_{SI}$ ,  $\vec{v}_{SI}$  are computed as described in Section E.1. The hyperbolic time  $\Delta t$  along that hyperbola from periapsis to  $r = r_{SI}$  is then determined from

$$S_f = p^{-1} (e^2 - 1)^{\frac{1}{2}} \sin v_{SI}$$

$$F = \log \left[ S_f + (S_f^2 + 1)^{\frac{1}{2}} \right]$$

$$\Delta t = \mu_P (eS_f - F) (v_{SI}^2 - 2\mu_P r_{SI}^{-1})^{-3}$$

4. Outer targeting - As explained above, all of the four basic targeting options require zero iterate trajectories that intersect the target planet sphere of influence. Since the point-to-point injection conditions will not always satisfy this requirement, some provision must be made to refine those conditions to obtain acceptable ones. An effective algorithm has been constructed to accomplish this.

When any trajectory has not encountered the target planet sphere of influence within a prescribed time, the closest approach conditions  $\vec{r}_{CA}$ ,  $\vec{v}_{CA}$ , are noted. An "artificial" sphere of influence is then constructed about the target planet, having a radius 1.2 times the value  $r_{CA}$  just noted. The trajectory will then intersect this artificial sphere of influence even with small perturbations in the injection velocity. The normal targeting procedure is now used under target option 2 ( $\vec{B} \cdot \hat{T}$ ,  $\vec{B} \cdot \hat{R}$ ,  $t_{SI}$ ) with the target conditions

$$\vec{B} \cdot \hat{T} = 0$$

$$\vec{B} \cdot \hat{R} = 0$$

$$t_{SI} = (t_{SI})_0 - \frac{1.2 r_{CA} - R_{SI}}{v_{CA}}$$

where  $(t_{SI})_0$  denotes the target time at the actual sphere of influence and  $R_{SI}$  is the actual sphere of influence radius. The injection conditions consistent with these "artificial" target conditions will in general yield a trajectory intersecting the actual sphere of influence of the target planet. The trajectory may then be targeted to the desired target conditions (see fig. 4).

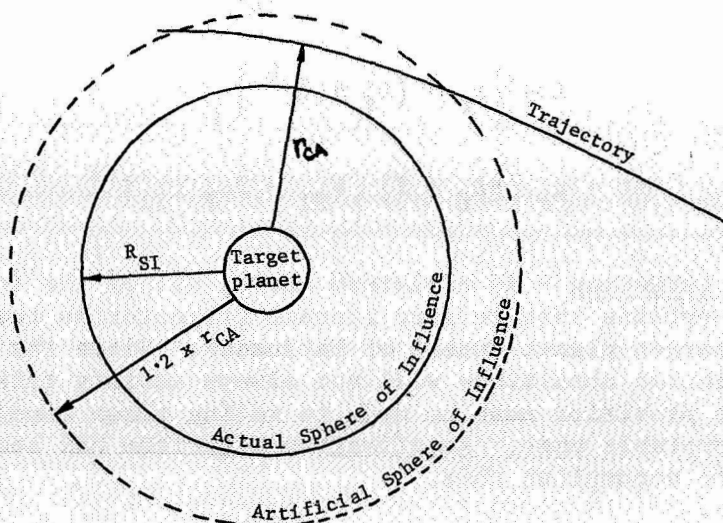


Figure 4. - Outer Targeting

## F. Specifics of Targeting Scheme

The standard numerical differencing scheme described in Section B has one glaring weakness: it is extremely time consuming. A simple numerical example illustrates the problem. The integration time for a reasonably accurate Mars trajectory (accuracy level =  $5 \times 10^{-6}$ , 2000 integration increments, five gravitational bodies) might be 30 sec (CDC 6500). The construction of the state transition matrix would require three integrations or 90 sec. Each iteration would, therefore, consume two minutes of computer time. Therefore, if two iterations in outer targeting and five iterations in normal targeting are required, the total computer time used in solving this problem is 14 minutes. For this reason, the standard method has been modified.

The modification is based on the premise that the characteristics of a low integration accuracy trajectory mirror those of a higher accuracy trajectory targeted to identical mission constraints. Thus, a state transition matrix computed about a targeted trajectory at a low accuracy level should remain valid for similar trajectories at higher accuracy levels. This assumption has been verified by experimentation.

The targeting scheme proceeds as follows. A trajectory is targeted to the desired target conditions at a low accuracy level, constructing the state transition matrix at each iteration as usual. The state transition matrix  $\phi^{(1)}$  evaluated about the targeted trajectory at this first accuracy level is stored. The injection velocity  $v^{(1)}$  of the targeted trajectory is then integrated forward at a higher accuracy level. Because of the change in the integration step size, the target conditions will not be realized. However, the matrix  $\phi^{(1)}$  may be used to predict an improved injection velocity  $v^{(2)}$  for that level. This process is repeated until the desired accuracy level is reached. Then the state transition matrix  $\phi^{(1)}$  is used iteratively to obtain a targeted trajectory at the final accuracy level.

The time savings is significant. Returning to the numerical example, suppose that intermediate accuracy levels requiring 250, 500, and 1000 integration increments were used. Seven iterations at the first level would use 140 sec (5 sec per integration x four integrations per iteration). One iteration at each of the

next two levels requires a total of 25 sec. Finally five iterations at the final accuracy level requires 150 sec (five integrations at 30 sec per integration). Thus, the total computer time of 6 minutes is less than half the previous time.

## V. ERROR ANALYSIS MODE - ANALYSIS

### A. Summary

The error analysis mode of STEAP is concerned with propagating uncertainties along n-body trajectories. No actual estimation is performed; the covariance matrices of state vector uncertainties are the primary output.

There are three main quantitative results that come from the error analysis mode, all of which are very important for trajectory design during preflight mission analysis. The first output is the orbit determination or navigation uncertainty at selected trajectory epochs. The processed covariance matrix of orbit determination uncertainty gives a probabilistic answer, for a specific reference trajectory, to the question "how well will the actual trajectory be known after optimal processing of the tracking information?" A second result obtained from the error analysis mode is equally important. Orbit determination uncertainties, although they are significant, do not by themselves answer all the pertinent questions related to mission success. Another question that must be answered is, "how close will the actual trajectory come to meeting the specified target conditions?"

To answer the second question satisfactorily, a midcourse correction process must be modeled in the error analysis mode. Between injection and the first midcourse correction, for example, tracking information is processed to reduce uncertainties associated with the estimated state vector. Because these uncertainties are independent in nonadaptive data processing schemes, they can be propagated in the error analysis mode. At the time of the first midcourse correction, the navigation uncertainty covariance matrix defines, probabilistically, the efficacy of the navigation algorithm. However, errors at injection, referenced to the nominal trajectory and related to the answer to the second key question, have propagated forward without any change due to the tracking data. Guidance corrections are made to ensure that the trajectory returns to the desired target conditions. The navigation uncertainty at the time of midcourse correction plus the error in performing the midcourse are the key contributors in determining, from the error analysis mode, how well the given guidance and navigation process will satisfy specified target conditions.

The error analysis mode also permits probabilistic determination of likely fuel budgets needed for interplanetary missions. Without performing any estimation, the most likely magnitudes of the midcourse correction magnitudes can be computed along with their variances. This computation permits the mission analyst to calculate reasonable fuel loading requirements that are critical in the design of an actual system.

Two matrix quantities are carried throughout the error analysis mode. One is the nominal or reference state vector, which is needed for many computations, and the second is the covariance matrix of navigation uncertainties associated with the state vector. The state vector is comprised of spacecraft position and velocity plus any augmentation parameters included in the analysis. The covariance matrix is a square, symmetric, positive definite matrix of associated uncertainties whose dimension corresponds to that of the state vector. It is implicit in the error analysis mode, when treating augmented states, that the underlying estimation algorithm is operating in a "solve for" mode; that is, the resulting uncertainties correspond to what occurs if the augmented parameters were actually estimated by the orbit determination process.

The computational operation of the error analysis mode may be separated into two distinct calculation procedures. The first of these is called the basic cycle and refers to the process of propagating uncertainties from one measurement to the next. A Kalman recursive filtering algorithm is used to process the measurement and compute the state vector associated covariance matrix that begins the next step in the basic cycle. Events refer to computations in the error analysis mode that are not simply propagations of the navigation uncertainty covariance matrix from one measurement to the next and subsequent optimal filtering of the new measurement. In the error analysis mode, three kinds of events are permitted.

The three events allowed in the error analysis mode are eigenvector events, prediction events, and guidance events. At an eigenvector event, the state vector associated covariance matrix is diagonalized to reveal geometric information about the size and orientation of the position and velocity navigation uncertainties. At a prediction event, the most recent covariance matrix is propagated forward to some critical trajectory epoch, usually a guidance correction time, to determine predicted orbit determination uncertainties in the absence of further measurements. When a guidance event occurs, a rather lengthy computational process

determines the likely magnitude of the guidance correction together with statistics of execution error based on an underlying physical model for the correction process.

Essentially then, the error analysis mode processes tracking information from various sources in the basic cycle and, at events, produces additional data not specifically related to the tracking sequence. This structure for the error analysis mode was designed to permit easy modification of the calculational procedure.

The next section of this chapter details the Kalman recursive estimation algorithm that is assumed, in the error analysis mode, to be the underlying orbit determination procedure. Sections C and D detail the manner in which the state transition and observation matrices, needed for the basic cycle, are computed in the error analysis mode. Section E presents the analytical equations used by the program at a guidance event to compute the most likely correction direction and magnitude as well as the execution error matrix. All three guidance policy options are discussed. The final two sections of the chapter explain the equations used for the other two events and the manner in which augmented states are handled by the error analysis mode.

## B. Recursive Estimation Algorithm

As was mentioned in the preceding section, the basic cycle of the error analysis mode is concerned with determining uncertainties associated with the state vector as a result of processing subsequent bits of tracking data. The recursive estimation algorithm refers to the computational procedure used in propagating the state vector covariance matrix of navigation uncertainties from one measurement to the next.

Let  $\bar{X}(t_k) = \bar{X}_k$  be the nominal state vector at the time  $t_k$  and define  $P_k^+$  as the navigation uncertainty covariance matrix after processing the measurement at the time  $t_k$ . Assume that the next measurement is made at the time  $t_{k+1}$  and that linear perturbations in that measurement are related to linear perturbations in the state vector by the observation matrix  $H_{k+1}$ . Assume further that the state transition matrix  $\Phi_{k+1,k}$  relating perturbations about the nominal trajectory at the times  $t_{k+1}$  and  $t_k$

has been calculated from the dynamic equations. If  $Q_{k+1,k}$  is a matrix of dynamic noise (or process noise) corrupting the dynamic equations over the given time interval, and  $R_{k+1}$  is the covariance matrix of a zero mean, Gaussian white noise process corrupting the measurement at  $t_{k+1}$ , then the standard Kalman recursive algorithm, which is assumed in the error analysis mode of STEAP, is given by

$$P_{k+1}^- = \Phi_{k+1,k} P_k^+ \Phi_{k+1,k}^T + Q_{k+1,k}$$

$$K_{k+1} = P_{k+1}^- H_{k+1}^T \left[ H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \right]^{-1}$$

$$P_{k+1}^+ = P_{k+1}^- - K_{k+1} H_{k+1} P_{k+1}^-$$

where  $P_{k+1}^-$  is the covariance matrix of navigation uncertainty at the time  $t_{k+1}$  prior to processing the new measurement,  $K_{k+1}$  is the Kalman gain matrix that optimally weights the dynamic and measurement information, and  $P_{k+1}^+$  is the navigation uncertainty matrix after the new measurement has been considered.

The above algorithm is well-known; details of its derivation may be found in reference 10. The new nominal state vector is obtained from the virtual mass n-body trajectory subroutine and is used in the computation of both  $\Phi_{k+1,k}$  and  $H_{k+1}$ , as will be explained in subsequent sections. The STEAP has modules that perform these calculations based on the kind of measurement taken at  $t_{k+1}$  and the method of state transition matrix generation being used for a particular run. The computation of  $Q_{k+1,k}$  within the program will be explained in the following paragraphs.

A phenomenon known as divergence in the filter has been noticed by many persons using the fundamental recursive algorithm for orbit determination or state vector estimation. The basic cause of divergence is modeling insufficiency and many separate categories of this insufficiency can be enumerated. The causes of the divergence problem and possible solutions to it are given in greater depth in the analytical discussion of the simulation mode. The purpose of including a dynamic noise matrix  $Q$  in the error



analysis mode is to check the effect of dynamic model insufficiency on the key outputs of the error analysis mode. Some dynamic or unmodeled noise always corrupts an interplanetary trajectory; what is interesting, from the point of view of the error analysis mode, is how the primary quantitative outputs are affected by various levels of dynamic noise.

The dynamic noise model used in the error analysis mode is somewhat arbitrary and its interpretation is difficult. Over any time interval  $\Delta t$  between measurements, the process noise matrix  $Q$  is computed from three input constants that remain the same throughout a trajectory run. These three constant inputs  $K_1$ ,  $K_2$ , and  $K_3$ , whose units are  $\text{km}^2/\text{sec}^4$ , roughly correspond to variances of assumed unmodeled accelerations. The dynamic noise matrix  $Q$  added over any interval  $\Delta t$  is diagonal and all zero except for the upper six diagonal terms corresponding to increased uncertainties in position and velocity. Specifically, if  $\Delta t$  is the interval between measurements, the six nonzero terms of  $Q$  are given by

$$\begin{aligned}
 Q_{11} &= \frac{1}{4}K_1\Delta t^4 \\
 Q_{22} &= \frac{1}{4}K_2\Delta t^4 \\
 Q_{33} &= \frac{1}{4}K_3\Delta t^4 \\
 Q_{44} &= K_1\Delta t^2 \\
 Q_{55} &= K_2\Delta t^2 \\
 Q_{66} &= K_3\Delta t^2
 \end{aligned} \tag{32}$$

Some explanation of this form for the dynamic noise is doubtless necessary. It was decided early in the design of the program that the physical interpretation of arbitrary dynamic noise must be made possible by relating the  $Q$  matrix, in some fashion, to unmodeled accelerations. Similarly, it appeared that the magnitude of the dynamic noise should be a function of the specific time interval over which it was added; in other words, the dynamic noise added when two days were between measurements should be greater than that added when only two hours separated the two measurements.

The first attempt to satisfy these two constraints resulted in the assumption that the unmodeled accelerations could be represented as biases with zero mean and variances  $K_1, K_2, K_3$ . Consider, for example, a vector random variable  $(\delta\ddot{X}, \delta\ddot{Y}, \delta\ddot{Z})^T$  with variances

$$\sigma_{\delta\ddot{X}}^2 = K_1 \quad \sigma_{\delta\ddot{Y}}^2 = K_2 \quad \sigma_{\delta\ddot{Z}}^2 = K_3 \quad (33)$$

and correlation coefficients set equal to zero. If these accelerations represent biases, then over any interval  $\Delta t$  they are related to position and velocity uncertainties through

$$\delta\dot{X} = \delta\ddot{X} (\Delta t); \quad \delta X = \frac{1}{2} (\delta\ddot{X}) (\Delta t)^2 \quad (34)$$

and similarly for the other components. Under this model for the dynamic noise, the  $Q$  matrix would be the same as that given in equation (32) except for the completely correlated off-diagonal terms resulting in

$$Q_{14} = \frac{1}{2} K_1 \Delta t^3, \quad Q_{25} = \frac{1}{2} K_2 \Delta t^3, \quad Q_{36} = \frac{1}{2} K_3 \Delta t^3 \quad (35)$$

Clearly if the unmodeled accelerations are indeed biases, the  $\delta X$  and  $\delta\dot{X}$  uncertainties due strictly to the dynamic noise must be completely correlated.

This initial model for the dynamic noise was unsatisfactory for two reasons. First, the resulting error analysis was forced to assume that the unmodeled acceleration was a constant bias throughout the trajectory as well as over each interval. The physics of the problem suggests that unmodeled accelerations are probably constant biases over short periods, but over an entire trajectory they probably vary considerably. Secondly, if the values for  $K_j$  are large enough for the dynamic noise to significantly affect the processed covariance matrices, their total correlation induces an unrealistically high correlation between the same terms in the resulting uncertainty matrices.

A more careful modeling of the stochastic process was discarded due to the arbitrary nature of the  $Q$  matrix. The dynamic noise matrix was chosen as in equation (32) because uncoupling the position and velocity uncertainties due to unmodeled accelerations retained a physical feel for the meaning of  $Q$  and permitted its computation to be viewed as a combination of random and bias error in the unmodeled accelerations.

### C. State Transition Matrices

This section presents the different techniques that are available for computing state transition matrices in STEAP. State transition matrices are used to relate linear perturbations about nominal state vectors at various specified times. If  $\bar{X}(t)$  is the nominal state vector at any time  $t$  and  $x(t)$  is a linear perturbation vector, then the matrix  $\Phi$  that satisfies

$$x(t_f) = \Phi(t_f, t_i) x(t_i) \quad (36)$$

between two selected epochs  $t_f$  and  $t_i$  is called the state transition matrix. As another way of defining the state transition matrix, consider the nominal state vector  $\bar{X}(t)$ , to be a function of some set of initial conditions  $\bar{X}(t_o)$ . For any arbitrary time  $t$ , a relationship exists such that

$$\bar{X}(t) = f[\bar{X}(t_o)] \quad (37)$$

Then taking differentials, one obtains

$$d\bar{X}(t) = x(t) = \frac{\partial f}{\partial \bar{X}} \left| \begin{array}{l} d\bar{X}(t_o) = \Phi(t, t_o) X_o \\ \text{evaluated} \\ \text{along} \\ \text{nominal} \end{array} \right. \quad (38)$$

From equation (38) and the linear difference equation (36), one can see that the state transition matrix  $\Phi$  is given by

$$\Phi(t_f, t_i) = \frac{\partial f}{\partial \bar{X}} \left| \begin{array}{l} \text{evaluated along} \\ \text{nominal} \end{array} \right. \quad (39)$$

State transition matrices are discussed in depth in reference 10.

Three methods for computing state transition matrices are available in STEAP. The first two methods are called analytical patched conic and analytical virtual mass. In the analytical methods, it is assumed that the interplanetary trajectory is a two-body conic section over a small time interval and that perturbations about the nominal trajectory can be related by using the basic analytical two-body matrizant. The third method for determining state transition matrices uses a numerical scheme that is often referred to as numerical differencing.

1. Analytical patched conic.— The basic idea in using an analytic patched conic state transition matrix is that over a small time interval of an interplanetary flight, the motion of a spacecraft is essentially a two-body conic section. Based on the foregoing assumption, Danby (ref. 6) has developed a set of general equations for determining the state transition matrices by the use of matrizants. The matrizant of two-body motion is used in STEAP for both analytical methods of computing state transition matrices. The basic fundamentals and equations of Danby's method will be presented here. Complete derivations are given in references 6 and 11.

Letting  $x(t)$  represent a column vector composed of position and velocity deviations at time  $t$ ,  $x(t_o)$  the same for time  $t_o$ , and  $g$  the deviation of a set of six geometrical elements, an equation that relates small deviations in position and velocity at two different times can be written as

$$x(t) = M(t)g = M(t)M^{-1}(t_o) x(t_o) \quad (40)$$

This equation is identical with the linear perturbation equation (36) except that now

$$\Phi(t, t_o) = M(t)M^{-1}(t_o) \quad (41)$$

The reference coordinate system considered here has the X-axis pointing toward periapsis for the conic, the Z-axis along the angular momentum vector, and Y forming the triad. Danby (ref. 6) calls this the "orbital reference system."

The geometrical orbital elements defined by  $g$  may be set in a column vector as,

$$g = \begin{bmatrix} \frac{\delta \ell_o + \eta \delta r}{n} \\ \frac{a \delta e}{h} \\ \delta p \\ \frac{\delta a}{2a} \\ \frac{ae \delta r}{h} \\ \delta q \end{bmatrix} \quad (42)$$

where  $\ell_o$  is the mean anomaly at an arbitrary epoch;  $a$  is the semimajor axis of the orbit;  $e$  is the eccentricity of the orbit;  $\delta p$ ,  $\delta q$ , and  $\delta r$  are infinitesimal rotations about the reference axis;  $h$  is the angular momentum per unit mass; and  $n$  is the mean motion of the orbit. The auxiliary parameter  $\eta$  is defined by  $(1 - e^2)^{1/2}$ . Avoiding the algebraic manipulations, the resultant  $M$  matrix as given by Danby (ref 6) has the following form,

$$M(t) = \begin{bmatrix} \dot{X} & Y\dot{X}-h & 0 & 2X-3t\dot{X} & Y\dot{Y} & 0 \\ \dot{Y} & -X\dot{X} & 0 & 2Y-3t\dot{Y} & -Y\dot{X}-2h & 0 \\ 0 & 0 & Y & 0 & 0 & -X \\ \ddot{X} & \dot{Y}\dot{X}+Y\ddot{X} & 0 & -\dot{X}-3t\ddot{X} & \dot{Y}^2+Y\ddot{Y} & 0 \\ \ddot{Y} & -\dot{X}^2-X\ddot{X} & 0 & -\dot{Y}-3t\ddot{Y} & -\dot{X}\dot{Y}-Y\ddot{X} & 0 \\ 0 & 0 & \dot{Y} & 0 & 0 & -\dot{X} \end{bmatrix} \quad (43)$$

where  $X$ ,  $Y$ , and  $Z$  are the components of position and velocity along the particular orbit,  $t$  is some specified epoch, and the accelerations are given by

$$\ddot{X} = \frac{-\mu X}{R^3}, \quad \ddot{Y} = \frac{-\mu Y}{R^3}, \quad \ddot{Z} = \frac{-\mu Z}{R^3}$$

$R$  is the magnitude of the position deviation represented by  $(X^2 + Y^2 + Z^2)^{1/2}$  and  $\mu$  is the gravitational constant of the dominant body used in the two-body approximation.

The inverse of the  $M$  matrix at the initial time  $t_0$  is given by

$$M^{-1}(t_0) = A \gamma M^T(t) \gamma^T \quad (44)$$

where  $A$  is a diagonal matrix of dimension  $6 \times 6$  and has diagonal components  $(a/\mu, a/\mu h, 1/h, a/\mu, a/\mu h, 1/h)$ .  $\gamma$  is given as the matrix

$$\gamma = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix},$$

with  $I$  being the identity matrix of appropriate dimension.

The state transition matrix that relates perturbations about some nominal state vector between two arbitrary times can now be determined by combining equations (43) and (44). The resulting matrix is referenced to the orbit plane coordinate system and thus, because all computations in STEAP are performed in the ecliptic frame, a rotation needs to be included so that

$$\Phi(t, t_0)_{\text{ecliptic}} = R \Phi(t, t_0)_{\text{orbit plane}} R^T \quad (45)$$

where  $R$  is the rotation matrix.

In using the foregoing method for analytical patched conic determination of the state transition matrices, an automatic check is made in the program to determine what sphere of influence the vehicle is in at the time of computation. The sphere of influence determines what gravitational mass and dominant body location will be used to compute the matrizant. It should be stressed that the

particular gravitational constant being used at the time of computing  $\Phi$  is chosen at the beginning of the time interval. In other words, if a check is made at  $t_1$  and the sphere of influence is that of the Sun and the trajectory at  $t_2$  is inside the sphere of influence of the target planet, then  $\mu_{\text{Sun}}$  will be used in the algorithm. No significant problems have resulted by using this approximate strategy, primarily because most of the time intervals are small when state transition matrices are computed near the spheres of influence.

The method of computing state transition matrices by the analytical patched conic technique is assumed in the program unless otherwise specified by the input.

2. Analytical virtual mass.— Computation of state transition matrices by the analytical virtual mass technique is similar to the patched conic method. The same general equations developed by Danby (ref. 6) are also used in determining state transition matrices using the virtual mass concept.

The virtual mass technique requires that the location and magnitude of the virtual mass, as calculated by the virtual mass subroutine, be stored for use in the computation of  $\Phi$ . Once the computational intervals and values for the location and magnitude of the virtual mass have been determined for the nominal trajectory these same quantities are used to generate the state transition matrix. Hence, after determining the nominal trajectory, the nominal state vector  $X(t)$  is available along with a set of values  $\vec{r}_v(t)$  and  $\mu_v(t)$  representing the position and magnitude of the virtual mass.

As mentioned previously, the equations for the two-body matrix are also employed in this second method of computing the state transition matrix. However, now the dominant body is assumed to be the effective force center. Recall that in the analytic patched conic method, a check was made to determine what sphere of influence the vehicle was in at the beginning of the time interval. In using virtual mass concepts to compute the state transition matrices, a sphere of influence check is avoided. When calling the state transition matrix module, the gravitational parameter of the virtual mass  $\mu_v$  is used instead of  $\mu$  of the primary attracting body. The location of the virtual mass is likewise used in the determination of  $\Phi$  under this method.

3. Numerical differencing.— The third method of computing state transition matrices in STEAP involves straightforward numerical differencing. In the other two methods, the partial derivative matrix  $\phi_{k+1,k}$  satisfying

$$x(t_{k+1}) = \frac{\partial X_{k+1}}{\partial X_k} \bigg|_{\substack{x(t_k) = \phi_{k+1,k} x(t_k) \\ \text{evaluated along} \\ \text{nominal}}}$$

was computed from analytic two-body approximations. Numerical differencing requires considerably more computer time to yield  $\phi_{k+1,k}$  from numerical considerations.

To demonstrate numerical differencing, let  $\bar{X}(t_k)$  and  $\bar{X}(t_{k+1})$  be nominal state vectors at the epochs  $t_k$  and  $t_{k+1}$  respectively. Next consider  $\bar{X}(t_k) + [dX_1, 0, 0, 0, 0, \dots]^T$ , obtained by adding a small perturbation factor to the first component of  $\bar{X}(t_k)$ . If the virtual mass subroutine is now used to propagate the new state vector until the time  $t_{k+1}$ , a new state vector  $X_1(t_{k+1})$  results. Defining the vector  $\Delta X_1$  by

$$\Delta X_1(t_{k+1}) = X_1(t_{k+1}) - \bar{X}(t_{k+1})$$

the first column of the state transition matrix  $\phi$  is then equal to

$$(\phi_{k+1,k})^1 = \frac{1}{dX_1} \Delta X_1(t_{k+1})$$

The process is continued by adding small perturbation factors to each component of  $\bar{X}(t_k)$  separately and using the resulting deviations about  $\bar{X}(t_{k+1})$  to build up the matrix  $\phi$  one column at a time. Implicit in this development for  $\phi$  is that the small perturbation factors added to each element of  $\bar{X}(t_k)$  must be of a size satisfying the linearity assumptions.



All three methods of state transition matrix computation can be used in STEAP. The first two analytic methods use much less computer time than numerical differencing.

#### D. Measurement Processing

The computational models used in the error analysis mode for relating tracking data to the basic nonaugmented state vector are presented in this section. The tracking module is responsible for computing at each navigation epoch an observation matrix  $H$  relating uncertainties in the measurements to uncertainties in the nominal state vector. Linear perturbation theory is employed, based upon analytic expressions relating the measurements to the nominal vehicle state. At some time  $t_k$ , the observed quantities can be related to the nominal state  $\bar{X}_k$  by,

$$\bar{Z}_k = h(\bar{X}_k) \quad (46)$$

The relationship between deviations in the measurements and deviations in the nominal state vector can be written by expanding equation (46) in a Taylor series about the nominal,

$$Z_k = \bar{Z}_k + z_k = h(\bar{X}_k + x_k) = h(\bar{X}_k) + \left. \frac{\partial h}{\partial \bar{X}_k} \right| x_k + 0(x_k^2) \quad (47)$$

where  $z_k$  represents deviations in the observations. By subtracting equation (46) from equation (47) and neglecting higher order terms  $0(x_k^2)$  a relationship between deviations in the measurements and deviations in the nominal state vector can be written as,

$$z_k = H_k x_k \quad (48)$$

where

$$H_k = \left. \frac{\partial h}{\partial \bar{X}_k} \right| \text{ evaluated along the nominal}$$

and  $H$  is an  $m \times n$  matrix relating the perturbations. For the present program  $H_k$  can be expressed as

$$H_k = \frac{\partial (D, \dot{D}, \alpha, \beta)}{\partial \bar{X}_k} \quad (49)$$

where  $D$  and  $\dot{D}$  are range and range rate, respectively. The angles  $\alpha$  and  $\beta$  are the star-planet angles and apparent planet diameter.

Three models are used in the tracking module, one a simple Earth-based tracking station (idealized) at the center of the Earth that measures range and range rate. The second model has three stations on a rotating spherical Earth that measure range and range rate. The last model provides the capability for on-board tracking by measuring three star-planet angles and apparent planet diameter.

1. Earth-based radar tracking.— The idealized station (center of the Earth) is a secondary model compared to the rest of the tracking models. It is a simple model that assumes that at each observation time, the idealized tracking station can look directly overhead at the vehicle. Let the heliocentric ecliptic position and velocity of the spacecraft at some time be designated by  $X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}$  and the heliocentric ecliptic coordinates of the Earth at the same time be  $X_E, Y_E, Z_E, \dot{X}_E, \dot{Y}_E, \dot{Z}_E$ . Then the slant range vectors from the tracking station to the spacecraft are,

$$\begin{aligned} \bar{X} &= X - X_E & \dot{\bar{X}} &= \dot{X} - \dot{X}_E \\ \bar{Y} &= Y - Y_E & \dot{\bar{Y}} &= \dot{Y} - \dot{Y}_E \\ \bar{Z} &= Z - Z_E & \dot{\bar{Z}} &= \dot{Z} - \dot{Z}_E \end{aligned} \quad (50)$$

The nominal range measurement is given by,

$$D = (\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2)^{1/2} \quad (51)$$

Differentiation of the range equation gives the following range rate expression,

$$\dot{D} = \frac{\dot{\overline{XX}} + \dot{\overline{YY}} + \dot{\overline{ZZ}}}{(\overline{X}^2 + \overline{Y}^2 + \overline{Z}^2)^{1/2}} = \frac{\dot{\overline{XX}} + \dot{\overline{YY}} + \dot{\overline{ZZ}}}{D} \quad (52)$$

The deviations in the range and range rate measurements are obtained by taking the first variation in equations (51) and (52) with the result,

$$\begin{aligned} \delta D &= \frac{\partial D}{\partial X} \delta X + \frac{\partial D}{\partial Y} \delta Y + \frac{\partial D}{\partial Z} \delta Z \\ \delta \dot{D} &= \frac{\partial \dot{D}}{\partial X} \delta X + \frac{\partial \dot{D}}{\partial Y} \delta Y + \frac{\partial \dot{D}}{\partial Z} \delta Z + \frac{\partial \dot{D}}{\partial \dot{X}} \delta \dot{X} + \frac{\partial \dot{D}}{\partial \dot{Y}} \delta \dot{Y} + \frac{\partial \dot{D}}{\partial \dot{Z}} \delta \dot{Z} \end{aligned} \quad (53)$$

Evaluation of the partials in equation (53) yields the following relations,

$$\begin{aligned} \frac{\partial D}{\partial X} &= \frac{\partial \dot{D}}{\partial \dot{X}} = \frac{\overline{X}}{D} \\ \frac{\partial D}{\partial Y} &= \frac{\partial \dot{D}}{\partial \dot{Y}} = \frac{\overline{Y}}{D} \\ \frac{\partial D}{\partial Z} &= \frac{\partial \dot{D}}{\partial \dot{Z}} = \frac{\overline{Z}}{D} \\ \frac{\partial \dot{D}}{\partial \dot{X}} &= \frac{\dot{\overline{X}}}{D} - \frac{\overline{X}\dot{D}}{D^2} \\ \frac{\partial \dot{D}}{\partial \dot{Y}} &= \frac{\dot{\overline{Y}}}{D} - \frac{\overline{Y}\dot{D}}{D^2} \\ \frac{\partial \dot{D}}{\partial \dot{Z}} &= \frac{\dot{\overline{Z}}}{D} - \frac{\overline{Z}\dot{D}}{D^2} \\ \frac{\partial D}{\partial \dot{X}} &= \frac{\partial D}{\partial \dot{Y}} = \frac{\partial D}{\partial \dot{Z}} = 0 \end{aligned} \quad (54)$$

The first model in the tracking module measures range rate from the idealized station, hence the observation matrix  $H_k$  for a nonaugmented state vector is given by

$$H_k = \begin{bmatrix} \frac{\partial \dot{D}}{\partial X} & \frac{\partial \dot{D}}{\partial Y} & \frac{\partial \dot{D}}{\partial Z} & \frac{\partial \dot{D}}{\partial \dot{X}} & \frac{\partial \dot{D}}{\partial \dot{Y}} & \frac{\partial \dot{D}}{\partial \dot{Z}} \end{bmatrix} \quad (55)$$

The second model gives an observation matrix with dimension  $2 \times 6$  relating range and range rate perturbations from the idealized station to perturbations in the basic state vector,

$$H_k = \begin{bmatrix} \frac{\partial D}{\partial X} & \frac{\partial D}{\partial Y} & \frac{\partial D}{\partial Z} & 0 & 0 & 0 \\ \frac{\partial \dot{D}}{\partial X} & \frac{\partial \dot{D}}{\partial Y} & \frac{\partial \dot{D}}{\partial Z} & \frac{\partial \dot{D}}{\partial \dot{X}} & \frac{\partial \dot{D}}{\partial \dot{Y}} & \frac{\partial \dot{D}}{\partial \dot{Z}} \end{bmatrix} \quad (56)$$

The partial derivatives, equation (54), are now substituted into equations (55) and (56) to complete the derivation of the required observation matrix.

The third and fourth models, which measure range rate and range-range rate, respectively, from a station on a spherical rotating Earth, are somewhat more complicated. Assuming that a station has some altitude  $h$ , latitude  $\theta$ , and longitude  $\phi$  as shown in figure 5, the geocentric equatorial coordinates of the station at epoch time  $T$  are given by,

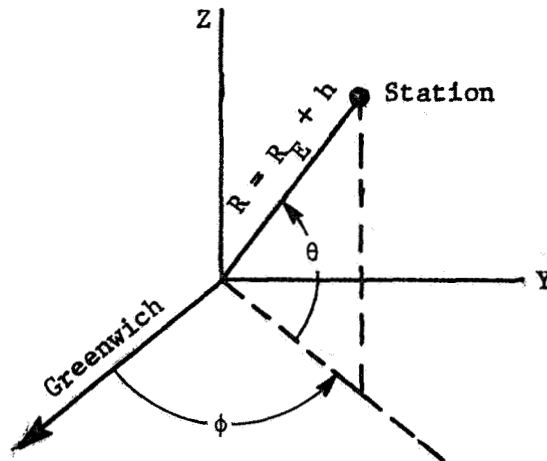


Figure 5.- Station Position

$$\begin{aligned}
X_T &= R \cos \theta \cos \phi \\
Y_T &= R \cos \theta \sin \phi \\
Z_T &= R \sin \theta
\end{aligned} \tag{57}$$

where  $R$  is the geocentric radius of the station given by  $R = R_E + h$  and  $R_E$  is the equatorial radius of the Earth. At any other time  $t$ , the position coordinates of the station are given by,

$$\begin{aligned}
X_T &= R \cos \theta \cos [\phi + \omega (t-T)] \\
Y_T &= R \cos \theta \sin [\phi + \omega (t-T)] \\
Z_T &= R \sin \theta
\end{aligned} \tag{58}$$

where  $\omega$  is the rotation rate of the Earth. The tracking station velocity is obtained by differentiating  $\vec{X}$  as,

$$\frac{d}{dt} (\vec{X}_T) = \dot{\vec{X}}_T = \begin{bmatrix} \dot{X}_T \\ \dot{Y}_T \\ \dot{Z}_T \end{bmatrix}$$

Thus,

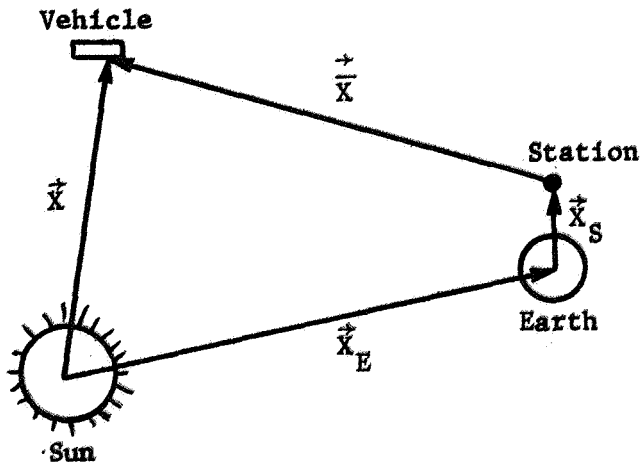
$$\begin{aligned}
\dot{X}_T &= -\omega R \cos \theta \sin [\phi + \omega (t-T)] \\
\dot{Y}_T &= \omega R \cos \theta \cos [\phi + \omega (t-T)] \\
\dot{Z}_T &= 0.
\end{aligned} \tag{59}$$

Equations (58) and (59) represent the geocentric equatorial coordinates of a tracking station on a spherical rotating Earth. Since most of the computational aspects of the error analysis mode are performed in the ecliptic system, the station coordinates must be referenced to the ecliptic system. The geocentric ecliptic

coordinates of the station are obtained by a rotation through the angle  $\epsilon$ , the obliquity of the ecliptic. The simple rotation yields the following time varying station coordinates in the ecliptic system,

$$\begin{aligned}
 X_S &= R \cos \theta \cos [\phi + \omega (t-T)] \\
 Y_S &= R \cos \theta \cos \epsilon \sin [\phi + \omega (t-T)] + R \sin \theta \sin \epsilon \\
 Z_S &= -R \cos \theta \sin \epsilon \sin [\phi + \omega (t-T)] + R \sin \theta \cos \epsilon \\
 \dot{X}_S &= -\omega R \cos \theta \sin [\phi + \omega (t-T)] \\
 \dot{Y}_S &= \omega R \cos \theta \cos \epsilon \cos [\phi + \omega (t-T)] \\
 \dot{Z}_S &= -\omega R \cos \theta \sin \epsilon \cos [\phi + \omega (t-T)]
 \end{aligned} \tag{60}$$

The slant range vector  $\rho$ , defined to be the position of the vehicle with respect to the ground station, can be determined by considering a vehicle in an orbit about the Sun where the vehicle motion is referenced to the heliocentric ecliptic system. The following sketch reflects the vector relationships,



where  $\vec{X}$  are the position coordinates of vehicle,  $\vec{X}_E$  are the position coordinates of Earth,  $\vec{X}_S$  are the position coordinates of station, and  $\vec{\rho}$  the vector from station to vehicle.

The range and range rate from the station to the vehicle are given by,

$$\rho = (\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2)^{1/2} \quad (61)$$

$$\dot{\rho} = \frac{\dot{\bar{X}}\bar{X} + \dot{\bar{Y}}\bar{Y} + \dot{\bar{Z}}\bar{Z}}{\rho} \quad (62)$$

where the position and velocity terms are,

$$\begin{aligned} \bar{X} &= X - X_E - X_S & \dot{\bar{X}} &= \dot{X} - \dot{X}_E - \dot{X}_S \\ \bar{Y} &= Y - Y_E - Y_S & \dot{\bar{Y}} &= \dot{Y} - \dot{Y}_E - \dot{Y}_S \\ \bar{Z} &= Z - Z_E - Z_S & \dot{\bar{Z}} &= \dot{Z} - \dot{Z}_E - \dot{Z}_S \end{aligned} \quad (63)$$

The station coordinates  $\vec{X}_S$  are represented by equation (60) and the coordinates of the Earth,  $\vec{X}_E$ , are determined from an ephemeris stored in the program. The vehicle position and velocity terms  $\vec{X}$ , are generated from the virtual mass subroutine.

The partial derivatives of range with respect to the state vector are,

$$\begin{bmatrix} \frac{\partial \rho}{\partial \bar{X}} \\ \frac{\partial \rho}{\partial \bar{Y}} \\ \frac{\partial \rho}{\partial \bar{Z}} \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} X - X_E - X_S \\ Y - Y_E - Y_S \\ Z - Z_E - Z_S \end{bmatrix} \quad (64)$$

$$\frac{\partial \rho}{\partial \dot{X}} = \frac{\partial \rho}{\partial \dot{Y}} = \frac{\partial \rho}{\partial \dot{Z}} = 0$$

and the corresponding derivatives of range rate are,

$$\begin{bmatrix} \frac{\partial \dot{\rho}}{\partial \dot{X}} \\ \frac{\partial \dot{\rho}}{\partial \dot{Y}} \\ \frac{\partial \dot{\rho}}{\partial \dot{Z}} \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}}{\rho} - \frac{\dot{X}\rho}{\rho^2} \\ \frac{\dot{Y}}{\rho} - \frac{\dot{Y}\rho}{\rho^2} \\ \frac{\dot{Z}}{\rho} - \frac{\dot{Z}\rho}{\rho^2} \end{bmatrix} = \begin{bmatrix} \frac{\dot{X} - \dot{X}_E - \dot{X}_S}{\rho} - \frac{\dot{\rho}(X - X_E - X_S)}{\rho^2} \\ \frac{\dot{Y} - \dot{Y}_E - \dot{Y}_S}{\rho} - \frac{\dot{\rho}(Y - Y_E - Y_S)}{\rho^2} \\ \frac{\dot{Z} - \dot{Z}_E - \dot{Z}_S}{\rho} - \frac{\dot{\rho}(Z - Z_E - Z_S)}{\rho^2} \end{bmatrix} \quad (65)$$

$$\begin{bmatrix} \frac{\partial \dot{\rho}}{\partial \dot{X}} \\ \frac{\partial \dot{\rho}}{\partial \dot{Y}} \\ \frac{\partial \dot{\rho}}{\partial \dot{Z}} \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} = \frac{1}{\rho} \begin{bmatrix} X - Y_E - X_S \\ Y - Y_E - Y_S \\ Z - Z_E - Z_S \end{bmatrix}$$

The required observation matrix  $H$  for model three (range rate) or model four (range and range rate) is obtained by assembling the above partial derivatives in accordance with the expressions,

$$H_k = \frac{\partial (\dot{\rho})}{\partial \vec{\dot{X}}} \text{ or } \frac{\partial (\rho, \dot{\rho})}{\partial \vec{\dot{X}}} \quad (66)$$

The observation matrices for these models can be computed by the program for any three tracking stations on a rotating Earth.

2. Onboard tracking - There are many types of onboard measurements that can be made by an interplanetary space vehicle for navigational purposes. Measurements such as Sun-planet angles, star-planet angles, star occultation, star-elevation angles, apparent planet diameter, and many others can be made by optical instruments. Battin discusses various on-board measurements in reference 9.



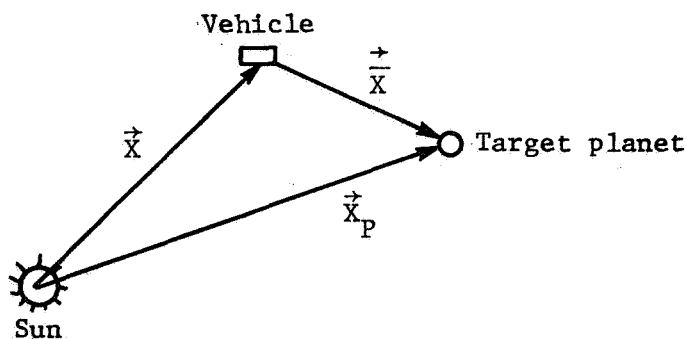
STEAP allows consideration of two types of onboard measurements, star-planet angles based on three reference stars, and apparent planet diameter. Star-planet angles can give an estimate of the vehicle's position by measuring the angle subtended at the spacecraft between the line of sight to a star and the line of sight to a near body (usually the target planet). The star's distance is assumed to be infinite so that its direction is independent of the point of observation. The second type of onboard measurement, apparent planet diameter, is useful in establishing position relative to the planet when the spacecraft is very close to the target.

The observation matrix  $H$  for star-planet angles and apparent planet diameter is determined by the same techniques as used in the range and range rate derivations. Equations are written in terms of the heliocentric position of the spacecraft and target planet and the appropriate partial derivatives are taken. The  $H$  matrix is given by,

$$H_k = \frac{\partial(\alpha_1, \alpha_2, \alpha_3, \beta)}{\partial \bar{X}} \quad (67)$$

where  $\alpha_1, \alpha_2, \alpha_3$  are the three star-planet angles and  $\beta$  is the apparent planet diameter. The observation matrix has dimension  $3 \times 6$  for star-planet angle measurements and  $1 \times 6$  for apparent planet diameter.

In determining the equations for the three star-planet angles consider the following vector diagram,



where  $\vec{X}$  is the heliocentric ecliptic vector to spacecraft and  $\vec{X}_P$  is the heliocentric ecliptic vector to target planet. The

position vector of the spacecraft with respect to the target planet is then,

$$\begin{bmatrix} \bar{X} \\ \bar{Y} \\ \bar{Z} \end{bmatrix} = \begin{bmatrix} X_P - X \\ Y_P - Y \\ Z_P - Z \end{bmatrix} \quad (68)$$

Let  $\vec{W}$  be a unit vector in the  $\frac{\vec{X}}{\rho}$  direction where

$$\vec{W} = \frac{\bar{X}}{\rho} \hat{i} + \frac{\bar{Y}}{\rho} \hat{j} + \frac{\bar{Z}}{\rho} \hat{k}$$

and

$$\rho = (\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2)^{1/2} \quad (69)$$

It is assumed that the direction cosines of the three reference stars are known and given by,

$$\text{Star No. 1} = (u_1, v_1, w_1)$$

$$\text{Star No. 2} = (u_2, v_2, w_2) \quad (70)$$

$$\text{Star No. 3} = (u_3, v_3, w_3)$$

where the direction cosines are referenced to the heliocentric ecliptic frame. If  $\alpha_n$  designates the star planet angles and the subscript  $n$  is the star number, the cosine of the angle measured is given by,

$$\cos \alpha_n = \frac{1}{\rho} [u_n \bar{X} + v_n \bar{Y} + w_n \bar{Z}]$$

or

$$\alpha_n = \cos^{-1} \left\{ \frac{1}{\rho} [u_n \bar{X} + v_n \bar{Y} + w_n \bar{Z}] \right\} \quad (71)$$

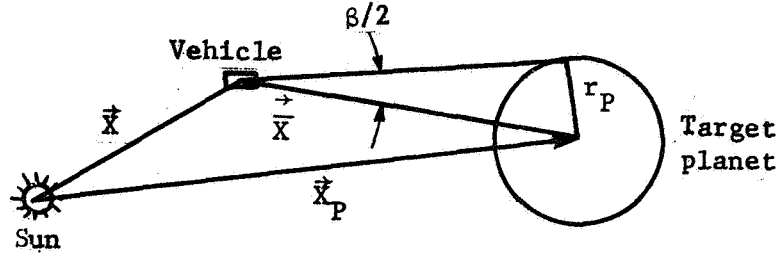
Taking derivatives of the star planet angles with respect to the state gives

$$\begin{aligned}
 \frac{\partial \alpha_n}{\partial \bar{X}} &= \frac{1}{\sin \alpha_n} \left[ \frac{u_n}{\rho} - \frac{\bar{X} \cos \alpha_n}{\rho^2} \right] \\
 \frac{\partial \alpha_n}{\partial \bar{Y}} &= \frac{1}{\sin \alpha_n} \left[ \frac{v_n}{\rho} - \frac{\bar{Y} \cos \alpha_n}{\rho^2} \right] \\
 \frac{\partial \alpha_n}{\partial \bar{Z}} &= \frac{1}{\sin \alpha_n} \left[ \frac{w_n}{\rho} - \frac{\bar{Z} \cos \alpha_n}{\rho^2} \right] \\
 \frac{\partial \alpha_n}{\partial \dot{\bar{X}}} &= \frac{\partial \alpha_n}{\partial \dot{\bar{Y}}} = \frac{\partial \alpha_n}{\partial \dot{\bar{Z}}} = 0 \\
 \sin \alpha_n &= \left[ 1 - \cos^2 \alpha_n \right]^{1/2}
 \end{aligned} \tag{72}$$

The required observation matrix  $H_k$  is obtained by insertion of the partial derivatives in equation (72) into the expression

$$H_k = \begin{bmatrix} \frac{\partial \alpha_1}{\partial \bar{X}} & \frac{\partial \alpha_1}{\partial \bar{Y}} & \frac{\partial \alpha_1}{\partial \bar{Z}} & \vdots \\ \frac{\partial \alpha_2}{\partial \bar{X}} & \frac{\partial \alpha_2}{\partial \bar{Y}} & \frac{\partial \alpha_2}{\partial \bar{Z}} & \vdots \\ \frac{\partial \alpha_3}{\partial \bar{X}} & \frac{\partial \alpha_3}{\partial \bar{Y}} & \frac{\partial \alpha_3}{\partial \bar{Z}} & \vdots \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{73}$$

To complete the derivation of onboard measurement types used in the program, the following paragraphs develop the necessary equations for apparent planet diameter. As mentioned previously, the apparent diameter is practical when the target planet is close enough for the measurement to be significant. Consider the following vector sketch,



For this figure,  $\vec{X}$  is the heliocentric ecliptic position of the spacecraft,  $\vec{X}_P$  is the heliocentric ecliptic coordinates of the target planet and  $r_P$  is the radius of the target planet. The vector  $\vec{X}$  is the position of the spacecraft with respect to the target planet and is computed as,

$$\begin{aligned}\bar{X} &= X_P - X \\ \bar{Y} &= Y_P - Y \\ \bar{Z} &= Z_P - Z\end{aligned}\tag{74}$$

The slant range of the vehicle is given by

$$\rho = (\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2)^{1/2}\tag{75}$$

The apparent angular diameter  $\beta$  is found from

$$\sin \frac{\beta}{2} = \frac{r_P}{\rho} = \frac{r_P}{(\bar{X}^2 + \bar{Y}^2 + \bar{Z}^2)^{1/2}}\tag{76}$$

Taking the first variation as before results in

$$\delta\beta = \frac{\partial\beta}{\partial X} \delta X + \frac{\partial\beta}{\partial Y} \delta Y + \frac{\partial\beta}{\partial Z} \delta Z$$

The corresponding partial derivatives are,

$$\begin{aligned}
 \frac{\partial \beta}{\partial \bar{X}} &= \frac{2 r_P \bar{X}}{\rho^2 \left[ \rho^2 - r_P^2 \right]^{1/2}} \\
 \frac{\partial \beta}{\partial \bar{Y}} &= \frac{2 r_P \bar{Y}}{\rho^2 \left[ \rho^2 - r_P^2 \right]^{1/2}} \\
 \frac{\partial \beta}{\partial \bar{Z}} &= \frac{2 r_P \bar{Z}}{\rho^2 \left[ \rho^2 - r_P^2 \right]^{1/2}} \\
 \frac{\partial \beta}{\partial \dot{\bar{X}}} &= \frac{\partial \beta}{\partial \dot{\bar{Y}}} = \frac{\partial \beta}{\partial \dot{\bar{Z}}} = 0
 \end{aligned}
 \tag{77}$$

The resulting observation matrix, which has dimension 1 x 6, is given by the expression,

$$H_k = \begin{bmatrix} \frac{\partial \beta}{\partial \bar{X}} & \frac{\partial \beta}{\partial \bar{Y}} & \frac{\partial \beta}{\partial \bar{Z}} & 0 & 0 & 0 \end{bmatrix}$$

where the partial derivatives are as shown in equation (77).

3. Measurement schedules.— The measurement schedules that are permitted by the error analysis mode of STEAP are completely arbitrary. Up to 500 measurements can be used for any specific run. The program contains a subroutine called SCHED that takes the set of input data and orders the measurements consecutively for processing. The input to the program, which is explained in the User's Manual (Volume I), is a set of variables that determines how many measurements will be used, what kind of measurements will be assumed by the process, and the frequency of each type of measurement.

## E. Midcourse Guidance Equations

This section is concerned with the computations made in the error analysis mode of STEAP that pertain to midcourse corrections made along an interplanetary trajectory. Previous sections have explained the analytic details associated with the basic cycle of the error analysis mode. The most complex computational event in the program is the guidance event. Accurate representations of guidance corrections and their influence on the overall error analysis are vital if the uncertainties resulting from using the error analysis mode are to be realistic.

Recall that the error analysis mode is essentially involved with propagating state vector uncertainties along some reference interplanetary orbit. The two quantities used for this purpose are the nominal state vector itself and the associated covariance matrix of navigation uncertainties. At time of a guidance correction in the error analysis mode, and because no estimation is actually performed, the nominal or reference trajectory is unchanged. Thus, the fundamental computation at a guidance event is the determination of the uncertainties in performing the maneuver; these uncertainties, which are a function of a number of quantities, are then added to the previous navigation covariance matrix for new processing in the tracking algorithm.

The computation of the maneuver uncertainty covariance matrix, called  $\tilde{Q}$  in the analysis to follow, is clearly dependent on the choice of guidance law or policy that is used for the corrections as well as the specific reference trajectory. The options available for the guidance policy will be explained in a subsequent section. Another factor affecting the calculation of  $\tilde{Q}$  is the modeling of the  $\Delta V$  vector at the time of the maneuver in the absence of any actual state vector estimation. The statistics of the process must be handled very carefully to produce realistic values for  $\Delta V$ . The third critical factor in determining  $\tilde{Q}$  is the execution error model itself.

1. Guidance policies.— The guidance policy assumed in the error analysis mode is responsible for the generation of what is known as the guidance matrix  $\Gamma$ . Assume that, for the purpose of defining  $\Gamma$ , an actual flight were being flown and that at some specified time for a midcourse maneuver, the orbit determination

procedure had produced the best estimate  $\hat{\delta X}$  of the actual state vector's deviation from some precomputed nominal that satisfied a set of specified target conditions. Under the assumption that the correction would be made using some kind of linear impulsive guidance scheme, the commanded correction  $\Delta V$  would be given by the matrix equation

$$\Delta V = \Gamma \hat{\delta X} \quad (78)$$

The matrix  $\Gamma$  is called the guidance matrix and is a function of the guidance policy being used.

Three kinds of guidance policies are modeled in the error analysis mode of STEAP. Each guidance policy represents a different way of correcting any deviations from the nominal trajectory to meet some set of targeting conditions. It should be stressed that for all three guidance policies, even though no estimation is performed in the error analysis mode, the computation of  $\Gamma$  can be made independent of the estimation algorithm. To determine  $\Gamma$  for any set guidance policy, all that is needed are the state transition matrices relating state vector uncertainties at the time of correction to uncertainties in the target conditions. A linearity assumption is clearly involved, hence the use of the phrase "linear impulsive guidance schemes."

The computational operation of the STEAP error analysis mode involves, at the time of a guidance maneuver, checking an input code to determine which of three policies is being used for the correction. The three guidance policy options are fixed-time-of-arrival (FTA), two-variable B-plane, and three-variable B-plane. The policy being assumed at the specific guidance correction is important only for the calculation of  $\Gamma$ ; once  $\Gamma$  has been computed, the rest of the calculations at a guidance event proceed independent of the kind of policy chosen.

If an FTA guidance policy has been specified, then the  $\Gamma$  matrix is calculated that would, for arbitrary linear perturbations about the reference trajectory at the maneuver time, result in a  $\Delta V$  which, if implemented perfectly, nulls position deviations about the reference trajectory at the time of closest approach to the target planet. Let  $\hat{\delta P}_c$  and  $\hat{\delta V}_c$  be estimates of arbitrary position and velocity deviations from the nominal trajectory at the correction time  $t_c$ . Let  $\Phi_{CA,c}$  be the 6 x 6 state transition matrix relating state vector uncertainties at

the time  $t_{CA}$  of closest approach to the target planet along the nominal trajectory to state vector uncertainties at the time  $t_c$  of the guidance maneuver. If  $\Phi_{CA,c}$  is partitioned into four  $3 \times 3$  matrices, then position and velocity errors at the target are given by

$$\begin{bmatrix} \delta P_{CA} \\ \delta V_{CA} \end{bmatrix} = \Phi_{CA,c} \begin{bmatrix} \delta \hat{P}_c \\ \delta \hat{V}_c \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix} \begin{bmatrix} \delta \hat{P}_c \\ \delta \hat{V}_c \end{bmatrix} \quad (79)$$

The desired correction vector  $\Delta V$  is the one that nulls the position deviation at the time of closest approach. Specifically,

$$\begin{bmatrix} 0 \\ dV_{CA} \end{bmatrix} = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix} \begin{bmatrix} \delta \hat{P}_c \\ \delta \hat{V}_c + \Delta V \end{bmatrix}, \quad (80)$$

which can be solved for  $\Delta V$  to produce

$$\Delta V = -\Phi_2^{-1} \Phi_1 \delta \hat{P}_c - \delta \hat{V}_c \quad (81)$$

and thus the guidance matrix  $\Gamma$  satisfying equation (78) is given in partitioned form as,

$$\Gamma = \begin{bmatrix} -\Phi_2^{-1} \Phi_1 & -I \end{bmatrix}.$$

Note that  $\Gamma$  is a  $3 \times 6$  matrix that has been separated into two square  $3 \times 3$  matrices. Also, from the above derivation,  $\Gamma$  itself is independent of the actual state vector estimate at the time of the guidance correction.

The calculation of  $\Gamma$  within STEAP, for the FTA guidance policy, is relatively straight forward. The state transition matrix  $\Phi_{CA,c}$  is computed as usual, by analytic or numerical means, and then properly partitioned and inserted to produce  $\Gamma$ .



For the three-variable B-plane guidance policy, the correction  $\Delta V$  is designed to null errors in the quantities  $B \cdot T$ ,  $B \cdot R$ , and time of piercing the sphere of influence. The B-plane geometry is explained in an earlier chapter. Again let  $\hat{\delta P}_c$  and  $\hat{\delta V}_c$  be estimates of arbitrary position and velocity derivations at the correction time. If  $\eta_{SI,c}$  is defined as a 3 x 6 linear variation matrix satisfying

$$\begin{bmatrix} \delta B \cdot T_{SI} \\ \delta B \cdot R_{SI} \\ \delta t_{SI} \end{bmatrix} = \eta_{SI,c} \hat{\delta X}_c = \begin{bmatrix} \eta_1 & \vdots & \eta_2 \end{bmatrix} \begin{bmatrix} \hat{\delta P}_c \\ \hat{\delta V}_c \end{bmatrix} \quad (82)$$

based upon the nominal trajectory, then the  $\Gamma$  matrix for the three-variable B-plane guidance policy may be written simply as

$$\Gamma = \begin{bmatrix} -\eta_2^{-1} & \eta_1 & \vdots & -I \end{bmatrix} \quad (83)$$

where the derivation follows the one for the FTA policy.

The actual computation of  $\Gamma$  for the three-variable B-plane policy is somewhat more complicated because the variation matrix must be determined numerically. This is because there are no good analytic formulas relating time at sphere of influence variations to earlier state vector variations.

The final guidance policy option is the two-variable B-plane policy. The implicit assumption in modeling  $\Gamma$  for this policy is that midcourse corrections are made that null only the deviations in  $B \cdot T$  and  $B \cdot R$  at the time the vehicle pierces the target planet sphere of influence. Errors in target time are considered acceptable. For this guidance policy, only two target conditions will be satisfied and there are three components of the  $\Delta V$  vector to be used for the midcourse maneuver. Thus, unlike the two previous policies, there is no unique solution for  $\Delta V$  and consequently, for the determination of  $\Gamma$ , an additional constraint must be defined. Let  $M$  be the 2 x 6 matrix, computed in the program in a straightforward manner, linearly relating

Perturbations about the nominal  $B \cdot T$  and  $B \cdot R$  to perturbations about the nominal state vector of position and velocity at the time of sphere of influence piercing,

$$\begin{bmatrix} \delta B \cdot T \\ \delta B \cdot R \end{bmatrix}_{SI} = M \delta X_{SI}$$

Then if  $\Phi_{SI,c}$  is the usual state transition matrix from the time at the sphere of influence to the time of the guidance correction, the variational relationship that will ultimately define  $\Gamma$  is given by

$$\begin{bmatrix} \delta B \cdot T \\ \delta B \cdot R \end{bmatrix}_{SI} = M \Phi_{SI,c} \delta \hat{X}_c = \eta \delta \hat{X}_c = A \delta \hat{P}_c + B \delta \hat{V}_c \quad (84)$$

where  $\eta$  now is a  $2 \times 6$  variation matrix, and  $A$  and  $B$  are  $2 \times 3$  partitions of  $\eta$  whose elements are obtainable from  $M$  and  $\Phi$  according to

$$A = (A_{\ell m}) \text{ where } A_{\ell m} = \sum_{k=1}^6 M_{\ell k} \Phi_{km} \quad (85)$$

$$B = (B_{\ell m}) \text{ where } B_{\ell m} = \sum_{k=1}^6 M_{\ell k} \Phi_{k(m+3)} \quad (86)$$

$$\ell = 1, 2; m = 1, 2, 3$$

From equation (84) it should be obvious that the two B-plane variations will be nulled by any midcourse correction vector  $\Delta V$  that satisfies the vector equation,

$$A \delta \hat{P}_c + B (\delta \hat{V}_c + \Delta V) = 0 \quad (87)$$

where, as before,  $\delta \hat{P}_c$  and  $\delta \hat{V}_c$  are estimates of arbitrary linear perturbations in the state vector position and velocity at the correction time. Rewriting the equation, any midcourse correction of the form

$$\Delta V = -B^T (BB^T)^{-1} A \delta \hat{P}_c - B^T (BB^T)^{-1} B \delta \hat{V}_c + u \quad (88)$$

where  $Bu = 0$ , will meet the two-variable B-plane targeting requirements. The value for  $u$  that minimizes the magnitude of  $\Delta V$  in equation (88) is  $u = 0$ , since  $Bu = 0$  implies that  $u$  is orthogonal, in the vector sense, to the remainder of the expression for  $\Delta V$ .

In implementing the two-variable B-plane guidance policy for the error analysis mode of STEAP, this minimum magnitude  $\Delta V$ , which corresponds to least fuel expenditure, is chosen. Thus the  $3 \times 6$  guidance matrix  $\Gamma$  for the two-variable policy is calculated, in partitioned form, as

$$\Gamma = \begin{bmatrix} -B^T (BB^T)^{-1} A & -B^T (BB^T)^{-1} B \end{bmatrix} \quad (89)$$

2. Modeling  $\Delta V$  without estimation.— The choice of guidance policy determines the way in which the  $\Gamma$  matrix is computed in the error analysis mode of STEAP. Once  $\Gamma$  has been calculated, the remainder of the guidance event algorithm is the same for all three guidance policies.

For the error analysis mode, the primary output of the guidance event algorithm is the execution error covariance matrix  $\tilde{Q}$  that will be added to the navigation uncertainty. Since a realistic  $\tilde{Q}$  is always a function of the commanded midcourse correction vector  $\Delta \vec{V}$ , some effective  $\Delta \vec{V}$ , based on the statistics of the flight, must be computed before  $\tilde{Q}$  can be obtained. This subsection of the report details the way in which the effective  $\Delta \vec{V}$  is calculated so that a realistic  $\tilde{Q}$  can be determined for further covariance matrix propagation.

The error analysis mode is essentially a preflight tool. Although it is known that guidance corrections, based on some guidance policy, will be applied at certain maneuver times according to the equation,

$$\Delta V = \Gamma \delta \hat{X} \quad (90)$$

in the absence of any estimated state vector  $\hat{\delta X}$ , all that is available for modeling  $\Delta V$  at the correction times are the likely statistics associated with the maneuver.

Since the injection covariance matrix assumed within the program is based on uncertainties about a nominal injection that will lead to specified target conditions, and since the orbit determination process is performed with an unbiased estimation algorithm, the a priori or preflight expected value for  $\hat{\delta X}$  at the correction time is zero. From the linearity of the expected value operator, it follows that the a priori expected value for the midcourse correction vector is also zero, even though there is only one injection, namely the nominal injection, that produces no midcourse corrections -- and this only occurs if the preflight trajectory computation had a perfect dynamic model.

Within the error analysis mode of STEAP, the execution error matrix  $\tilde{Q}$  is based upon a priori information related to the most likely magnitude of  $\Delta V$ , given by  $E[\Delta V]$ , and the direction of the assumed or effective  $\Delta V$  is then determined as the most likely direction, given that some correction will be made. The vector quantity " $E[\Delta V]$ " that appears in the following equations should be treated, therefore, as the effective midcourse correction vector used to determine realistic values for the execution error matrix  $\tilde{Q}$  in the absence of any estimation.

Define " $E[\Delta V_j]$ " as the effective midcourse correction vector to be used at the time of the  $j^{\text{th}}$  correction for computing the execution error matrix  $\tilde{Q}_j$ . Let  $P_{j-1}^+$  be the covariance matrix of orbit determination uncertainty -- combining navigation and execution uncertainties after the  $(j-1)^{\text{st}}$  correction -- just after the previous maneuver. For  $j=1$ ,  $P_0$  is the injection covariance matrix. If  $\Phi_{j,j-1}$  is the state transition matrix relating linear perturbations about the nominal at the times of the  $j^{\text{th}}$  and  $(j-1)^{\text{st}}$  correction, then, assuming a "correct" guidance policy at  $t_{j-1}$ , the statistical deviations of the actual trajectory from a reference trajectory that meets the specified targeting conditions are given, prior to the  $j^{\text{th}}$  correction, by the covariance matrix

$$P_j^- = \phi_{j,j-1} P_{j-1}^+ \phi_{j,j-1}^T + Q_{j,j-1} \quad (91)$$

where  $Q_{j,j-1}$  is the process noise matrix between corrections. Continuing the above reasoning, the a priori statistics on the midcourse correction vector  $\Delta \vec{V}_j$  at the time of the  $j^{\text{th}}$  correction may be calculated as

$$S_j = E \begin{bmatrix} \Delta V_j & \Delta V_j^T \end{bmatrix} = \Gamma_j P_j^- \Gamma_j^T \quad (92)$$

where  $\Gamma_j$  is the guidance matrix at the  $j^{\text{th}}$  correction. Since " $E[\Delta V_j]$ ", is critically based on the covariance matrix  $S_j$ , it is worthwhile to examine its origin in detail.

Before the flight, the injection is described in terms of uncertainties about some nominal by the injection covariance  $P_0$ . Since an arbitrary linear perturbation about the nominal injection is related to perturbations about the nominal state vector at the first correction time through the state transition matrix  $\phi_{1,0}$ , the covariance matrix  $P_1^-$ , where

$$P_1^- = \phi_{1,0} P_0 \phi_{1,0}^T + Q_{1,0} \quad (93)$$

represents the a priori likely dispersions of the actual trajectory from the nominal trajectory at the time of the first correction. Since

$$\Delta V_1 = \Gamma_1 \hat{\delta X}_1 \quad (94)$$

no matter which value for  $\hat{\delta X}_1$  actually occurs, the a priori statistical description of the first midcourse correction vector is that it has mean zero and covariance matrix  $S_1$ , where

$$S_1 = \Gamma_1 P_1^- \Gamma_1^T \quad (95)$$

Based on the effective  $\Delta V$ , the covariance matrix of navigation uncertainty  $P_1^+$ , given by

$$P_1^+ = P_1^- + \begin{bmatrix} 0 & 0 \\ 0 & Q_1 \end{bmatrix} \quad (96)$$

is used to replace the injection covariance  $P_0$  and thus the a priori statistical description of the second midcourse correction is contained in the covariance matrix  $S_2$ , where

$$S_2 = \Gamma_2 \left[ \Phi_{2,1} P_1^+ \Phi_{2,1}^T + Q_{2,1} \right] \Gamma_2^T. \quad (97)$$

The general process continues, from correction to correction, according to equations (91) and (92).

The covariance matrix  $S_j$  thus represents the set of all values for  $\Delta V_j$  that would occur, with some assigned probabilities, if the actual flight were being flown. For example, if  $P_0$  is the injection covariance matrix describing the statistics of injection uncertainty, then  $S_1$  is the covariance matrix that describes all ensemble correction vectors  $\Delta V_1$  resulting from injection dispersions defined by  $P_0$ .

Before any information about an actual flight is available, the midcourse correction vector  $\Delta V_j$  at the  $j^{\text{th}}$  correction is a random variable, with mean zero and associated covariance matrix  $S_j$ . Define  $\rho_j$  as the scalar variable given by

$$\rho_j = E \left[ \|\Delta V_j\| \right] \quad (98)$$

In this equation  $\rho_j$  represents the a priori most likely magnitude of the  $j^{\text{th}}$  midcourse correction and hence is very important for fuel sizing. Hoffman and Young (ref. 12) have shown that a good approximation for the quantity  $\rho_j$  is given by

$$\rho_j \approx \sqrt{\frac{2A}{\pi}} \left( 1 + \frac{B (\pi-2)}{A^2 \sqrt{5.4}} \right) \quad (99)$$

where

$$A = \text{trace } S_j,$$

$$B = \lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_2 \lambda_3, \quad (100)$$

and  $\lambda_1, \lambda_2, \lambda_3$  are the eigenvalues of the covariance matrix  $S_j$ .

The statistical variance of the magnitude of the  $j^{\text{th}}$  midcourse correction, also derived in approximate form in reference 12, is given by the equation,

$$\sigma_{|\Delta V_j|}^2 = \text{trace } S_j - \rho_j^2 \quad (101)$$

Both these values are computed in the guidance event algorithm of the error analysis mode.

From the point of view of the error analysis mode, the most likely magnitude for the  $j^{\text{th}}$  correction  $\Delta V_j$  is simply  $\rho_j$ . To complete the specification of the effective midcourse correction vector " $E[\Delta V_j]$ ", all that remains is to determine its most likely direction.

Let  $\lambda_1, \lambda_2$ , and  $\lambda_3$  be the eigenvalues of the covariance matrix  $S_j$  associated, a priori, with the  $j^{\text{th}}$  midcourse maneuver. Define  $\alpha_1, \alpha_2$ , and  $\alpha_3$  as the eigenvectors related to the three eigenvalues. It can be shown that, under the assumption that some correction takes place, the most likely direction for the midcourse maneuver, defined probabilistically, is the direction of the eigenvector associated with the maximum eigenvalue of  $S_j$ . Define  $\vec{\alpha}$  as this eigenvector associated with the maximum eigenvalue. The effective midcourse correction vector associated with the  $j^{\text{th}}$  correction is then given by

$${}^{\text{E}}[\Delta V_j] = \frac{\rho_j \vec{\alpha}}{|\vec{\alpha}|}$$

This vector has magnitude  $\rho_j$ , the most likely magnitude for  $\Delta V_j$  based upon the a priori statistics, and is in the direction of the maximum probability correction under the assumption that some correction is made.

It should be stressed that the computation of the effective midcourse correction vector  ${}^{\text{E}}[\Delta V_j]$ , within the error analysis mode, is only an artifice to permit a realistic, a priori computation of the execution error matrix  $\tilde{Q}_j$ . The nominal trajectory returned to the basic cycle is not affected by the computation. However, the calculated information concerning likely magnitudes and directions for the maneuvers is critical for fuel sizing studies.

3. Execution error modeling.— This subsection of the report treats the analytical way in which STEAP computes the execution error matrix  $\tilde{Q}_j$  associated with a specified midcourse correction vector  $\Delta V_j$ . In the error analysis mode, the input to this portion of the guidance event algorithm is the effective midcourse correction vector  ${}^{\text{E}}[\Delta V_j]$  defined in the preceding subsection. The problem may be stated simply: given a midcourse correction vector  $\Delta V$ , determine an associated covariance matrix of execution errors from some physical model identifying relevant error sources.

Within the STEAP guidance event algorithm, four independent error sources are modeled to be used in the determination of the execution error matrix  $\tilde{Q}$ . The first error source is called the proportionality error and is in the direction of the midcourse correction vector  $\Delta V$  with magnitude determined by the proportionality factor  $k$ . A second error source, in the direction of  $\Delta V$  but independent of its magnitude, is the resolution error  $s$  that corresponds to a thrust tailoff error from the midcourse engines. Two pointing errors defined in terms of angles  $\delta\alpha$  and  $\delta\beta$  complete the error model for the midcourse maneuver.



Let  $\delta\Delta V$  be the error vector at the time of the midcourse maneuver. From the above description, a general vector equation for the maneuver error is given by

$$\delta\Delta V = k\Delta V + s \frac{\Delta V}{|\Delta V|} + \delta\Delta V_{\text{pointing}} \quad (102)$$

where  $\delta\Delta V_{\text{pointing}}$  will be described later. The execution error covariance matrix  $\tilde{Q}$  that is needed at the time of the midcourse correction is simply

$$\tilde{Q} = E [\delta\Delta V \delta\Delta V^T] \quad (103)$$

Assume now that  $\delta\Delta V_{\text{pointing}}$  results from two angular pointing errors,  $\delta\alpha$  and  $\delta\beta$ . For purposes of unique specification, assume that  $\delta\alpha$  is a pointing error angle measured in a plane parallel to the ecliptic plane and along a vector orthogonal to the midcourse correction vector  $\Delta V$ . If  $\delta\Delta V_1$  is the velocity error due to the angular pointing error  $\delta\alpha$  and  $\hat{i}, \hat{j}, \hat{k}$  form the unit triad in the heliocentric ecliptic system, then, for small angles  $\delta\alpha$ ,

$$\delta\Delta V_1 = \rho\delta\alpha \left[ \frac{\Delta V_Y}{(\Delta V_X^2 + \Delta V_Y^2)^{1/2}} \hat{i} - \frac{\Delta V_X}{(\Delta V_X^2 + \Delta V_Y^2)^{1/2}} \hat{j} \right] \quad (104)$$

where  $\Delta V_X$  and  $\Delta V_Y$  are the X and Y components in the heliocentric ecliptic system of the midcourse correction vector  $\Delta V$  and  $\rho$  is the magnitude of  $\Delta V$ . Note that the velocity error  $\delta\Delta V_1$  resulting from  $\delta\alpha$  has components only in a plane parallel to the ecliptic.

The second pointing angle  $\delta\beta$  defines a velocity error  $\delta\Delta V_2$  that is orthogonal to both  $\delta\Delta V_1$  and the midcourse correction vector  $\Delta V$ . Again for small angles  $\delta\beta$ , the velocity error resulting from this pointing error, referenced to the heliocentric ecliptic system, is given by

$$\delta\Delta V_2 = \frac{\Delta V_X \Delta V_Z \delta\beta}{(\Delta V_X^2 + \Delta V_Y^2)^{1/2}} \hat{i} + \frac{\Delta V_Y \Delta V_Z \delta\beta}{(\Delta V_X^2 + \Delta V_Y^2)^{1/2}} \hat{j} - \delta\beta (\Delta V_X^2 + \Delta V_Y^2)^{1/2} \hat{k} \quad (105)$$

From these equations it is clear that the vector set  $\Delta V$ ,  $\delta\Delta V_1$  and  $\delta\Delta V_2$  satisfies the mutual orthogonality imposed by the model. The complete description of the execution error vector  $\delta\Delta V$  may then be written in heliocentric ecliptic coordinates as

$$\begin{aligned}\delta\Delta V = & \left[ \left( k + \frac{s}{\rho} \right) \Delta V_X + \frac{\rho \Delta V_Y \delta\alpha + \Delta V_X \Delta V_Z \delta\beta}{u} \right] \hat{i} \\ & + \left[ \left( k + \frac{s}{\rho} \right) \Delta V_Y + \frac{\Delta V_Y \Delta V_Z \delta\beta - \rho \Delta V_X \delta\alpha}{u} \right] \hat{j} \\ & + \left[ \left( k + \frac{s}{\rho} \right) \Delta V_Z - u \delta\beta \right] \hat{k}\end{aligned}\quad (106)$$

where  $\Delta V_X$ ,  $\Delta V_Y$ , and  $\Delta V_Z$  are the heliocentric ecliptic coordinates of the midcourse correction;  $\hat{i}$ ,  $\hat{j}$ ,  $\hat{k}$  are unit vectors in the X, Y, and Z directions;  $\rho$  is the magnitude of  $\Delta V$ ;  $k$ ,  $s$ ,  $\delta\alpha$ ,  $\delta\beta$  are the four independent error sources to be treated as random variables; and  $u$  is an intermediate variable defined by

$$u = \left( \Delta V_X^2 + \Delta V_Y^2 \right)^{1/2} \quad (107)$$

From the above expression for  $\delta\Delta V$ , the execution error covariance matrix  $\tilde{Q}$  can be computed, in terms of its elements, as

$$\begin{aligned}\tilde{Q}_{11} &= \Delta V_X^2 \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + \frac{\Delta V_Y^2 \rho^2 \sigma_{\delta\alpha}^2}{u^2} + \frac{\Delta V_X^2 \Delta V_Z^2 \sigma_{\delta\beta}^2}{u^2} \\ \tilde{Q}_{12} = \tilde{Q}_{21} &= \Delta V_X \Delta V_Y \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] - \frac{\rho^2 \Delta V_X \Delta V_Y \sigma_{\delta\alpha}^2}{u^2} + \frac{\Delta V_X \Delta V_Y \Delta V_Z^2 \sigma_{\delta\beta}^2}{u^2} \\ \tilde{Q}_{13} = \tilde{Q}_{31} &= \Delta V_X \Delta V_Z \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] - \Delta V_X \Delta V_Z \sigma_{\delta\beta}^2 \\ \tilde{Q}_{22} &= \Delta V_Y^2 \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + \frac{\Delta V_X^2 \rho^2 \sigma_{\delta\alpha}^2}{u^2} + \frac{\Delta V_Y^2 \Delta V_Z^2 \sigma_{\delta\beta}^2}{\Delta V_X^2 + \Delta V_Y^2} \\ \tilde{Q}_{23} = \tilde{Q}_{32} &= \Delta V_Y \Delta V_Z \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] - \Delta V_Y \Delta V_Z \sigma_{\delta\beta}^2 \\ \tilde{Q}_{33} &= \Delta V_Z^2 \left[ \sigma_k^2 + \frac{\sigma_s^2}{\rho^2} \right] + u^2 \sigma_{\delta\beta}^2\end{aligned}\quad (108)$$

The computation of the execution error covariance matrix is dependent upon the assumed perfect midcourse  $\Delta V$  with components  $\Delta V_X$ ,  $\Delta V_Y$ ,  $\Delta V_Z$ , and the variances (zero means are assumed) of the four independent error sources. In the above equations, it is assumed that the errors have no cross-correlations and that  $\sigma_k^2$ ,  $\sigma_s^2$ ,  $\sigma_{\delta\alpha}^2$ , and  $\sigma_{\delta\beta}^2$  are the variances of the proportionality, resolution, and two pointing errors respectively.

Recall that, in the error analysis mode, no actual state vector estimation is performed, and hence no specific midcourse corrections are ever calculated. The components  $\Delta V_X$ ,  $\Delta V_Y$ ,  $\Delta V_Z$  used in the above expression for computation of  $Q$  come from the effective midcourse correction vector " $E[\Delta V]$ " discussed earlier.

4. Additional computations.— Some additional quantities of mission analysis importance are also calculated in the guidance event algorithm of the error analysis mode of STEAP. It has already been pointed out that fuel sizing information results from the computation of the most likely direction and magnitude for the midcourse maneuver  $\Delta V$ . The propagation of navigation uncertainties by the basic cycle results in orbit determination accuracies in the neighborhood of the target planet. The third key piece of mission analysis data, which tells how well a defined navigation and guidance process will meet specified target conditions, is provided at a guidance event.

Before any midcourse corrections, the statistical deviations at the target about the nominal or reference target conditions may be found by propagating the injection covariance all the way to the target. Let  $\eta_o$  be a  $3 \times 6$  variation matrix relating linear perturbations in three targeting conditions to linear injection perturbations. Then, if  $P_o$  is the injection covariance, the statistical description of target condition deviations about the nominal due to injection errors is given by the matrix  $W_o^+$ , where

$$W_o^+ = \eta_o P_o \eta_o^T \quad (109)$$

In the absence of dynamic noise, the matrix  $W_o^+$  represents statistically how far the ensemble package of injections defined by  $P_o$  would miss the specified target conditions. Tracking

information taken after the injection can lower the inaccuracies in the answer to the question "where is the spacecraft?" However, it is only at guidance corrections (again in the absence of unmodeled accelerations) in the error analysis mode that there is any change in the probabilistic dispersions about the desired target conditions.

At each guidance correction in the error analysis mode, two covariance matrices are computed that are related to likely target condition dispersions about the nominal. For the  $j^{\text{th}}$  midcourse correction, the matrix  $W_j^-$  represents target condition dispersions about the nominal if no correction were to take place at that time. It is called the covariance matrix of uncertainties in the target conditions before the correction. Similarly, once the correction has been modeled and an execution error covariance  $\tilde{Q}_j$  obtained, another matrix  $W_j^+$  is computed that represents target condition dispersions after the correction. Both these matrices would play a critical role in any midcourse optimization scheme designed to choose the times for the corrections.

Within the program,  $W_j^-$  is computed as

$$W_j^- = \eta_j \left[ \Phi_{j,j-1} P_{j-1}^+ \Phi_{j,j-1}^T + Q_{j,j-1} \right] \eta_j^T \quad (110)$$

where  $\eta_j$  is the linear variation matrix relating target condition deviations to perturbations in the state vector at the time of the  $j^{\text{th}}$  correction,  $P_{j-1}^+$  is the navigation uncertainty covariance after the  $j-1^{\text{st}}$  correction that contains the execution error covariance  $\tilde{Q}_{j-1}$  from the  $j-1^{\text{st}}$  correction, and the other two quantities are the usual state transition matrix and process noise matrix. The covariance matrix  $W_j^+$ , defining uncertainties in the target conditions after the  $j^{\text{th}}$  midcourse maneuver, is given by

$$W_j^+ = \eta_j P_j^+ \eta_j^T = \eta_j \left( P_j^- + \begin{bmatrix} 0 & 0 \\ 0 & \tilde{Q}_j \end{bmatrix} \right) \eta_j^T \quad (111)$$

where  $P_j^-$  is the orbit determination uncertainty covariance matrix just prior to the correction and  $\tilde{Q}_j$  is the execution error covariance matrix for the  $j^{\text{th}}$  correction.

It is illustrative to demonstrate how the successive computation of the  $W$  matrices is helpful from a mission analysis point of view. Assume first that the process noise matrix  $Q$  is always so insignificant that it can be neglected. The matrix  $W_0^+$  given in equation (109) represents the propagation of ensemble injection errors into target condition deviations. In the absence of dynamic noise,  $W_1^-$  is given by

$$W_1^- = \eta_1 \phi_{1,0} P_0 \phi_{1,0}^T \eta_1^T = W_0^+ \quad (112)$$

$$\text{since} \quad \eta_1 \phi_{1,0} = \eta_0 \quad (113)$$

As would be expected, there have been no guidance corrections to bring the trajectory back to the specified target conditions and thus the target condition uncertainties are unchanged.

An effective midcourse maneuver " $E \begin{bmatrix} \Delta V_1 \end{bmatrix}$ " is next calculated and its associated execution error matrix  $\tilde{Q}_1$  is determined. The covariance matrix  $P_1^-$  is the navigation uncertainty before the maneuver. Then

$$W_1^+ = \eta_1 \left( P_1^- + \begin{bmatrix} 0 & 0 \\ - & - \\ 0 & \tilde{Q}_1 \end{bmatrix} \right) \eta_1^T \quad (114)$$

Under the implicit assumption that if the spacecraft trajectory uses perfectly known  $\begin{pmatrix} P_1^- = 0 \end{pmatrix}$  and the execution error were small  $\begin{pmatrix} \tilde{Q}_1 = 0 \end{pmatrix}$ , then the guidance policy would result in a corrected trajectory satisfying the desired target conditions.

Continuing to the second correction,  $W_2^- = W_1^+$  in the absence of dynamic noise and then

$$W_2^+ = \eta_2 \left( P_2^- + \begin{bmatrix} 0 & 0 \\ - & - \\ 0 & Q_2 \end{bmatrix} \right) \eta_2^T \quad (115)$$

tells the analyst how much the second correction is likely to reduce the target condition dispersions.

In a later portion of this report, test runs with the error analysis mode are presented. Most of the data discussed are concerned with the likely magnitudes of the corrections, orbit determination uncertainties along the trajectory, and the matrices  $W_j^-$  and  $W_j^+$  which define probabilistically the efficacy of the navigation and guidance process.

It should be mentioned here that STEAP has an additional option available to the error analysis mode for computing the execution error matrix  $\tilde{Q}_j$  without using the effective midcourse velocity vector " $E[\Delta V_j]$ ". Under the same formulation for  $\delta \Delta V$  given by equation (106), if the components  $\Delta V_X$ ,  $\Delta V_Y$  and  $\Delta V_Z$  are also treated (as they are, in reality) as random variables in determining  $\tilde{Q}_j$  an approximate solution for  $\tilde{Q}_j$  can be obtained by using the matrix  $S_j$  describing the statistics of the correction vector  $\Delta V_j$ . The validity of the result is difficult to determine physically, however, and the mathematics are sufficiently nonrigorous to preclude judgement on the accuracy of the resulting approximation.

#### F. Eigenvector and Prediction Events

The two other computational events in the error analysis mode are eigenvector and prediction events. At an eigenvector event, the covariance matrix of orbit determination uncertainty is operated on to produce geometric information about the size and orientation of the navigation uncertainties. At a prediction event, the navigation uncertainties are propagated forward, assuming no additional measurements, to determine uncertainties at critical trajectory epochs.

The computations at an eigenvector event are straightforward. Define  $t_k$  as the time of the last processed measurement before the eigenvector event and let  $\bar{X}_k$  and  $P_k^+$  be, respectively, the nominal trajectory and the orbit determination uncertainty covariance matrix after processing the measurement at  $t_k$ . If  $t_e$  is the time of the eigenvector event, then  $\bar{X}_e$  the nominal state vector at  $t_e$ , is computed from the virtual mass trajectory subroutine. The navigation uncertainty covariance matrix at  $t_e$  defined by  $P_e$  is given by

$$P_e = \phi_{e,k} P_k^+ \phi_{e,k}^T + Q_{e,k} \quad (116)$$

where  $\phi_{e,k}$  is the state transition matrix between  $t_k$  and  $t_e$  and  $Q_{e,k}$  is the process noise matrix. At an eigenvector event, the quantities  $\bar{X}_e$  and  $P_e$  are first computed and then stored for use by the basic cycle of the error analysis mode after the eigenvector calculations are finished.

In its general form, considering all possible augmented state vector options,  $P_e$  is a square matrix whose dimension is given by the dimension of the augmented state vector. The first six components of the state vector are always position and velocity, and hence the upper left 6 x 6 of  $P_e$  is the covariance matrix of uncertainties associated with position and velocity. Define by  $\tilde{P}_e$  the 3 x 3 partition of  $P_e$  containing information about the positional navigation uncertainties; define by  $\hat{P}_e$  the 3 x 3 partition of  $P_e$  related to velocity navigation uncertainties.

At an eigenvector event, both  $\tilde{P}_e$  and  $\hat{P}_e$  are diagonalized in a standard way to produce the position and velocity eigenvalues and the associated eigenvectors. In addition, the 36 hyperellipsoids of uncertainty, in both the position and the velocity space, are computed to show the size and geometric orientation of the navigation uncertainties. These ellipsoids are then projected onto

each of the two-dimensional planes to show additional geometric information. All these calculations are straightforward and will be discussed only briefly in a general form.

Let  $P$  be a  $3 \times 3$  covariance matrix associated with a zero mean, three dimensional vector random variable  $X$ . If  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are the eigenvalues of  $P$  and  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  are the associated eigenvectors, then the physical meaning of statistical deviations about the mean can best be viewed in terms of an ellipsoid in the three-dimensional space whose principal axes are in the direction of the three eigenvectors and have length equal to the magnitude of the eigenvalues. The  $3\sigma$  ellipsoids computed from  $P$  result in a geometrical surface of points of equal probability whose interior contains over 95% of the probabilistic events. The principal axis associated with the minimum eigenvalue defines the direction of minimum uncertainty; the axis associated with the maximum eigenvalue defines the direction of maximum uncertainty.

At a prediction event, the nominal trajectory  $\bar{X}_e$  and associated navigation uncertainty covariance matrix  $P_e$  are first calculated just as at an eigenvector event. Now define  $t_p$  as a time to which the prediction is being made. For example, suppose that orbit determination uncertainties are desired at launch plus five days as a result of tracking for four days after launch. Then  $t_e$  is launch plus four days and  $t_p$  is launch plus five days. A typical use of the prediction event information might be for loading a guidance command maneuver. If a correction were to be made at launch plus five days and were to be loaded at launch plus four days, then the key covariance matrix of navigation uncertainty would be the predicted covariance at five days, given only the results of tracking up to four days.

Assume that the prediction event occurs at  $t_e$  and that the prediction time is  $t_p$ . Then the navigation uncertainty covariance matrix at  $t_p$ , given the measurements taken before  $t_e$ , is simply

$$P_p = \Phi_{p,e} P_e \Phi_{p,e}^T + Q_{p,e} \quad (117)$$



where, as before,  $\Phi_{p,e}$  and  $Q_{p,e}$  are the usual state transition matrix and process noise matrix respectively. Within the prediction event algorithm of the error analysis mode, the resulting covariance matrix  $P_p$  at the prediction time is also diagonalized to produce eigenvector, eigenvalue, and hyperellipsoid information. Thus, by superimposing this geometrical information about  $P_p$  for different prediction event times  $t_e$ , one can observe the effect of additional tracking on predicted navigation uncertainties.

Within the error analysis mode of STEAP, if the prediction time  $t_p$  is close to the time  $t_{SI}$  that the nominal trajectory pierces the sphere of influence of the target planet, then additional computations are made. Suppose  $t_p = t_{SI}$  and that the  $2 \times 6$  matrix  $M$ , defined earlier, relates linear perturbations about the nominal B-plane parameters  $B \cdot T$  and  $B \cdot R$  to linear perturbations about the nominal state vector at the time  $t_{SI}$ . In this situation the program also computes predicted navigation uncertainties in terms of the B-plane parameters by calculating  $\tilde{P}$ , a  $2 \times 2$  matrix, where

$$\tilde{P} = M P_p M^T \quad (118)$$

and then diagonalizes  $\tilde{P}$  to show its orientation in the impact parameter or B-plane.

One additional computation of note takes place at both an eigenvector and a prediction event. Before eigenvalue information is determined, the entire correlation coefficient matrix  $\tilde{R}$ , whose elements are given by

$$\tilde{R}_{ij} = \frac{P_{i,j}}{\sqrt{P_{i,i} P_{j,j}}} \quad (119)$$

is computed. This matrix is also square, has dimensionality equal to the dimension of the state vector, and yields the correlation coefficients between various element uncertainties in the state vector. When any off-diagonal element of  $\tilde{R}$  becomes close to unity in magnitude, the underlying system becomes less and less observable. A condition coefficient of unit magnitude relating

any two components indicates that the estimation algorithm cannot uniquely determine those two elements separately. The concepts of observability and partial observability will be discussed in greater detail in the analytical presentation of the simulation mode.

#### G. State Vector Augmentation

Both the error analysis and simulation modes of STEAP permit the use of augmented state vectors. In reference 10 a detailed derivation of the required recursive estimation equations is given when all the augmented parameters are constants. The basic Kalman filtering algorithm remains unchanged, although the inclusion of more elements in the state vector does increase the dimensionality of the problem.

To illustrate the handling of augmented states within both the error analysis and simulation modes of STEAP, consider an augmented state vector  $\delta Z$ , assumed to be linear perturbations, that can be partitioned as

$$\delta Z = \begin{bmatrix} \delta X \\ \delta P_1 \\ \delta P_2 \end{bmatrix} \quad (120)$$

where  $\delta X$  is a  $6 \times 1$  vector of position and velocity deviations (the nonaugmented state),  $\delta P_1$  is a  $n_1 \times 1$  vector of perturbations in parameters that appear in the dynamic equations, and  $\delta P_2$  is a  $n_2 \times 1$  vector of perturbations in constants that

appear only in the measurement equations. If  $\hat{\delta Z}_k$  is the best

estimate of this augmented state at  $t_k$ ,  $P_k^+$  is the  $(6 + n_1 + n_2) \times (6 + n_1 + n_2)$  processed covariance matrix of orbit determination uncertainty at  $t_k$ , and  $\Delta Y_{k+1}$  is the measured deviation of an observation made at  $t_{k+1}$  from some nominal measurement based upon the nominal trajectory, then the full, augmented recursive algorithm for both the error analysis and simulation modes is given by

$$\begin{aligned}
P_{k+1}^- &= \Psi_{k+1,k} P_k^+ \Psi_{k+1,k}^T + \Lambda_{k+1,k} \\
\Sigma_{k+1} &= M_{k+1} P_{k+1}^- M_{k+1}^T + R_{k+1} \\
K_{k+1} &= P_{k+1}^- M_{k+1}^T \Sigma_{k+1}^{-1} \\
P_{k+1}^+ &= P_{k+1}^- - K_{k+1} M_{k+1} P_{k+1}^- \\
\hat{\delta Z}_{k+1} &= \Psi_{k+1,k} \hat{\delta Z}_k + K_{k+1} \left[ \Delta Y_{k+1} - M_{k+1} \Psi_{k+1,k} \hat{\delta Z}_k \right],
\end{aligned} \tag{121}$$

where  $\Psi_{k+1,k}$  is the state transition matrix of dimension  $(6 + n_1 + n_2) \times (6 + n_1 + n_2)$ ;  $\Lambda_{k+1,k}$  is a square process noise matrix of augmented dimensions;  $M_{k+1}$  is the observation matrix of dimension  $r \times (6 + n_1 + n_2)$  where  $r$  is the dimensionality of the measurement;  $R_{k+1}$  is the same measurement noise matrix as before; and  $K_{k+1}$ , the Kalman gain matrix, now has dimension  $(6 + n_1 + n_2) \times r$ . Under the reasonable assumption that there is no dynamic or process noise corrupting the augmented parameters, then  $\Lambda_{k+1,k}$  may be partitioned as,

$$\Lambda_{k+1,k} = \begin{bmatrix} Q_{k+1,k} & 0 \\ 0 & 0 \end{bmatrix} \tag{122}$$

the only new or additional computations required for the augmented state, other than changes in matrix dimensionality, are in the calculation of the observation matrix  $M_{k+1}$  and the state transition matrix  $\Psi_{k+1,k}$ .

Recall that the observation matrix  $M_{k+1}$  relates linear perturbations about the nominal measurement at  $t_{k+1}$  to linear perturbations in the state vector at the same time. Let  $\delta y_{k+1}$  be a measurement perturbation of dimensionality  $r \times 1$  at the time  $t_{k+1}$ . Then, in partitioned form, the measurement equation may be written

$$\delta y_{k+1} = M_{k+1} \delta Z_{k+1} = \begin{bmatrix} H_{k+1} & 0 & B_{k+1} \end{bmatrix} \begin{bmatrix} \delta X_{k+1} \\ \delta P_1 \\ \delta P_2 \end{bmatrix} \quad (123)$$

where  $H_{k+1}$  is just the  $r \times 6$  observation matrix for the non-augmented state, 0 is a  $r \times n_1$  matrix of zeros since the dynamic augmented parameters are not related to the measurements explicitly, and  $B_{k+1}$  is a  $r \times n_2$  matrix defining changes in the observation due to changes in the augmented parameters that appear in the measurement equations.

Similarly, if the defining equations for the state transition matrix  $\Psi_{k+1,k}$  are written in partitioned form as

$$\begin{bmatrix} \delta X_{k+1} \\ \delta P_1 \\ \delta P_2 \end{bmatrix} = \delta Z_{k+1} = \Psi_{k+1,k} \delta Z_k = \begin{bmatrix} \phi_{k+1,k} & \theta_{k+1,k} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \delta X_k \\ \delta P_1 \\ \delta P_2 \end{bmatrix} \quad (124)$$

the matrix  $\phi_{k+1,k}$  is recognized as the nonaugmented, standard  $6 \times 6$  state transition matrix. Since the augmented parameters are all constants, identity matrices of dimensionality  $n_1$  and  $n_2$  appear in the partitioned expression for  $\Psi_{k+1,k}$ . Furthermore, since perturbations  $\delta P_2$  in the measurement equation parameters do not affect anything else in the state vector between measurements, and errors in position and velocity have no influence on perturbations in the augmented dynamic constants, the only additional computations required for the augmented state transition matrix involve the determination of the  $6 \times n_1$  matrix  $\theta_{k+1,k}$ . This matrix describes the linear perturbations in position and velocity, over the given time interval, due to perturbations in the dynamic constants.

Essentially then, other than changing the dimensionality of the matrices involved, the only additional calculations required, for either estimation or error propagation with an augmented state, are those used for the determination of the matrices  $B_{k+1}$  and  $\theta_{k+1,k}$  defined above.

1. Options.— Ten augmented state vector options are currently available in STEAP for both the error analysis and simulation modes. An input code value, IAUG, specifies the state vector to be used in the computations. Spacecraft position and velocity are always included in the state vector. The following table summarizes the augmented state vector options.

IAUG	Augmented Parameters	Dimension of state	Description
1	None	6	Standard spacecraft position and velocity
2	$dR_1, d\theta_1, d\phi_1$	9	Includes geocentric radius, latitude, and longitude biases for first station location
3	$du_s, du_{tp}$	8	Gravitational biases for sun and target planet
4	$dD_1, d\dot{D}_1, da_1$ $da_2, da_3, d\beta$	12	Measurement biases - range and range-rate from first station, three star-planet angles, apparent diameter
5	$da_p, de_p, di_p$	9	Target planet ephemeris biases in conic section elements
6	$dR_1, d\theta_1, d\phi_1$ $dR_2, d\theta_2, d\phi_2$ $dR_3, d\theta_3, d\phi_3$	15	Geometric radius, latitude and longitude biases from all three station locations
7	$dR_1, d\theta_1, d\phi_1$ $du_s, du_{tp}$	11	Combination of IAUG = 2 and IAUG = 3
8	$dR_1, d\theta_1, d\phi_1$ $dD_1, d\dot{D}_1, da_1$ $da_2, da_3, d\beta$	15	Combination of IAUG = 2 and IAUG = 4
9	$du_s, du_{tp}, da_p$ $de_p, di_p$	11	Includes all dynamic constant biases
10	$dD_1, d\dot{D}_1, da_1$ $da_2, da_3, d\beta$ $da_p, de_p, di_p$	15	Combination of IAUG = 4 and IAUG = 5
11	$dR_1, d\theta_1, d\phi_1, du_s,$ $du_{tp}, dD_1, d\dot{D}_1, da_1$ $da_2, da_3, d\beta$	17	Combination of IAUG = 2, IAUG = 3, and IAUG = 4

From the preceding table it is clear that the maximum dimensionality of the propagated covariance matrix in the error analysis mode is  $17 \times 17$ . Notice that target planet ephemeris uncertainties are treated as biases in the conic section elements  $a$ ,  $e$ ,  $i$ . This is in keeping with the ephemeris routine used by the n-body trajectory scheme that computes the position and velocity of the planets from mean conic section orbital elements.

2. Observation matrices.— A full description of the augmented observation matrix  $M_{k+1}$  given in equation (123) requires only the additional computation of the matrix  $B_{k+1}$  relating linear perturbations about the nominal measurements to linear perturbations about the nominal values for the augmented parameters appearing in the measurement equations. A total of 15 possible parameters, including nine station location biases and six measurement biases, may be treated under one of the augmented state options as augmented measurement parameters. Each of these biases is treated by the program as mean zero Gaussian random variables with specified variances.

Assume first that augmented state option  $IAUG = 4$ , whose state vector has 12 dimensions, is being run. Partitioning the measurement equation,

$$\delta y_{k+1} = M_{k+1} \delta Z_{k+1} = \begin{bmatrix} H_{k+1} & | & B_{k+1} \end{bmatrix} \begin{bmatrix} \delta X_{k+1} \\ \delta D_1 \\ \delta \dot{D}_1 \\ \delta \alpha_1 \\ \delta \alpha_2 \\ \delta \alpha_3 \\ \delta \beta \end{bmatrix} \quad (125)$$

where  $\delta X_{k+1}$  is a  $6 \times 1$  vector of perturbations in position and velocity, it is easy to see that the determination of the matrix  $B_{k+1}$  is dependent only on the measurement type being taken. If the particular measurement being made is range and range-rate, then the range and range-rate biases are simply additive to produce

$$B_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Similarly, if the measurements being processed are star-planet angles, then the remainder of the observation matrix  $M_{k+1}$  is given by

$$B_{k+1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The  $H_{k+1}$  partition of  $M_{k+1}$  is always computed in the same way as it was for the nonaugmented state.

Handling the station location biases is more difficult computationally. The only measurements affected by these biases are the measurements of range-rate and range. Assume, for illustration, that  $IAUG = 6$  and that at the time  $t_{k+1}$  a measurement of range and range-rate from station 2 is taken. The partitioned form of the measurement equation is now given by

$$\begin{bmatrix} \delta D_2 \\ \delta \dot{D}_2 \end{bmatrix}_{k+1} = M_{k+1} \delta Z_{k+1} = \begin{bmatrix} H_{k+1} & 0 & B_{k+1} & 0 \end{bmatrix} \begin{bmatrix} \delta X_{k+1} \\ \delta R_1, \delta \theta_1, \delta \phi_1 \\ \delta R_2 \\ \delta \theta_2 \\ \delta \phi_2 \\ \delta R_3, \delta \theta_3, \delta \phi_3 \end{bmatrix} \quad (126)$$

The partitioned zeros of  $M_{k+1}$  are of dimension  $2 \times 3$  and occur **because** biases in station locations 1 and 3 do not affect a range and range-rate measurement made from station 2. Since  $H_{k+1}$  is already known from standard, nonaugmented procedures, all that

remains to complete the calculation of  $M_{k+1}$  is the specification of the  $2 \times 3$  partitioned matrix  $B_{k+1}$  that relates linear perturbations in the range and range-rate measurements to linear perturbations in the station 2 geocentric coordinates.

Recall from Section D of this chapter that the range and range-rate measurements are functions of the geocentric station location coordinates. Mathematically,

$$\begin{aligned} D_2 &= f_1(X, R_2, \theta_2, \phi_2, t_{k+1}) \\ \dot{D}_2 &= f_2(X, R_2, \theta_2, \phi_2, t_{k+1}) \end{aligned} \quad (127)$$

Expanding in Taylor series about the nominal values, the matrix  $B_{k+1}$  can be written as

$$B_{k+1} = \begin{matrix} (2 \times 3) & \begin{bmatrix} \frac{\partial D_2}{\partial R_2} & \frac{\partial D_2}{\partial \theta_2} & \frac{\partial D_2}{\partial \phi_2} \\ \frac{\partial \dot{D}_2}{\partial R_2} & \frac{\partial \dot{D}_2}{\partial \theta_2} & \frac{\partial \dot{D}_2}{\partial \phi_2} \end{bmatrix} \end{matrix} \quad \begin{matrix} \text{evaluated along the} \\ \text{nominal at } t_{k+1} \end{matrix} \quad (128)$$

The analytic partial derivatives defining  $B_{k+1}$  may be easily computed from the equations given in Section D.

For any of the possible augmented state vector options, the matrix  $B_{k+1}$  is computed according to the above procedures, depending on which measurement parameters are included in the augmented state. The STEAP error analysis and simulation modes then put together the proper observation matrix  $M_{k+1}$  from  $H_{k+1}$ ,  $B_{k+1}$ , and the particular augmented state vector being used.

3. State transition matrices.— To compute the state transition matrix  $\Psi_{k+1,k}$  for the augmented state vector, it is clear from equation (124) that the only additional calculations involved are associated with the generation of the submatrix  $\theta_{k+1,k}$ . This



submatrix relates linear perturbations in position and velocity at the time  $t_{k+1}$  to linear perturbations in the dynamic constants that affect the trajectory over the time interval  $\Delta t = t_{k+1} - t_k$ .

Within the error analysis and simulation modes of STEAP, five dynamic parameters may be added to the state vector. These five are the gravitational constants of the sun and target planet,  $\mu_s$  and  $\mu_{tp}$ , and three target planet ephemeris biases,  $a_p$ ,  $e_p$ , and  $i_p$ . Assume, again for illustration, that  $IAUG = 9$ . Then the partitioned equation for the state transition matrix  $\Psi_{k+1,k}$  is given by

$$\begin{bmatrix} \delta X_{k+1} \\ \delta \mu_s \\ \delta \mu_{tp} \\ \delta a_p \\ \delta e_p \\ \delta i_p \end{bmatrix} = \begin{bmatrix} \delta X_{k+1} \\ \delta P \end{bmatrix} = \delta Z_{k+1} = \Psi_{k+1,k} \delta Z_k = \begin{bmatrix} \phi_{k+1,k} & \theta_{k+1,k} \\ 0 & I \end{bmatrix} \begin{bmatrix} \delta X_k \\ \delta P \end{bmatrix} \quad (129)$$

where  $\phi_{k+1,k}$  is the 6 x 6 state transition matrix for the non-augmented state and  $\theta_{k+1,k}$  is the 6 x 5 submatrix whose computation completes the specification of the entire state transition matrix  $\Psi_{k+1,k}$ .

To explain the genesis of  $\theta_{k+1,k}$  recall that the solution  $X_{k+1}$  to the spacecraft equations of motion at the time  $t_{k+1}$  can be written, in symbolic form, as

$$X_{k+1} = f(X_k, P, t) \quad (130)$$

Let  $\bar{X}_k$  be the nominal position and velocity at  $t_k$  and let  $\bar{P}$  represent the nominal values for the dynamic parameters. If

$\bar{X}_{k+1}$  is the nominal solution to the equations of motion; that is,

$$\bar{X}_{k+1} = f(\bar{X}_k, \bar{P}, t) \quad (131)$$

then linear variations around that nominal, allowing for perturbations in both  $\bar{X}_k$  and  $\bar{P}$ , may be written as

$$\delta X_{k+1} = \left. \frac{\partial f}{\partial \bar{X}_k} \right|_{\substack{\text{evaluated along} \\ \text{the nominal}}} \delta X_k + \left. \frac{\partial f}{\partial \bar{P}} \right|_{\substack{\text{evaluated along} \\ \text{the nominal}}} \delta P \quad (132)$$

The quantity  $\frac{\partial f}{\partial \bar{P}}$  in the above expression may be recognized as the state transition submatrix  $\theta_{k+1,k}$ .

Within STEAP, the matrices  $\theta_{k+1,k}$  are always computed by numerical differencing. To illustrate the calculation of one column of  $\theta_{k+1,k}$ , let  $\bar{X}_{k+1}$  and  $\bar{X}_k$  be nominal positions and velocities at  $t_{k+1}$  and  $t_k$  respectively. These nominal values result from the assumption that the gravitational constant of the Sun is given by  $\mu_s$ . Now set  $\mu_s = \mu_s + \Delta\mu_s$  and perform the n-body trajectory computation over the interval  $[t_k, t_{k+1}]$ . A new value for the spacecraft position and velocity, call it  $\tilde{X}_{k+1}$ , results. Define the vector  $\Delta X = \tilde{X}_{k+1} - \bar{X}_{k+1}$ . Divide each component of  $\Delta X$  by the perturbation magnitude  $\Delta\mu_s$  and the resultant vector forms the column of  $\theta_{k+1,k}$  that associates linear perturbations in position and velocity with linear perturbations in the gravitational constant for the Sun.

When  $\text{IAUG} = 9$ , five separate integrations must be used to build up the five columns of  $\theta_{k+1,k}$ . Each of the five dynamic constant biases is varied separately and the trajectory is computed again. These numerical quantities should be good approximations to the analytic partials defined by equation (132).

The computation of the state transition and observation matrices for augmented states is, as has been shown, relatively straight forward. However, storing these calculated values in proper arrays, depending on the augmented state option is a complex bookkeeping problem. It is hoped that a future version of STEAP will permit arbitrary augmentation (specification of any combination of additional parameters) with many more possible augmented parameters.

## VI. SIMULATION MODE - ANALYSIS

There is an essential difference between the philosophies governing the error analysis and simulation modes of STEAP. The error analysis mode is primarily a preflight mission analysis tool that gives information related to uncertainties about some specified nominal trajectory. By contrast, the simulation mode is designed for a detailed analysis of the orbit determination procedure and its efficacy in the presence of a host of possible anomalies. The error analysis mode might be used to determine the nominal trajectory design for a specific mission; the simulation mode then "flies" the mission, within the computer, and can provide invaluable information for mission operations.

The computational structure of the simulation mode is similar to that of the error analysis mode. There is a basic cycle in which subsequent measurements are processed consecutively and there are events where calculations not specifically related to the measurement-processing cycle are made. The first section of this chapter outlines in detail the equations and logic used within the basic cycle of the simulation mode. The second section treats the most complex event, namely the guidance event where midcourse corrections are simulated. In the third section the other events are presented, including the quasilinear filtering event that is included for its possible divergence prevention abilities. The fourth section includes a discussion of the problems of divergence and nonobservability that can plague an orbit determination procedure.

The computations themselves, within the simulation mode, are not any more difficult than in the error analysis mode. There are, however, many more of them and in the discussion to follow some of the most important features of the simulation mode may be discussed. Chapter VII Section D of this volume, which contains comments on the results of several actual runs with the simulation mode, will help clarify and explain the many possible uses of this portion of STEAP.

### A. The Basic Cycle

Recall that in the error analysis mode only two vector quantities, the nominal state vector  $\bar{X}$  and its associated covariance matrix of orbit determination uncertainty  $P$ , were carried along

through each step in the basic cycle. Within the simulation mode there are five key quantities carried from step to step in the basic cycle. Let  $t_k$  and  $t_{k+1}$  be the times of the  $k^{\text{th}}$  and  $k + 1^{\text{st}}$  measurements, with no events between them. Available in the simulation mode are the five quantities:

- 1)  $\bar{X}(t_k) = \bar{X}_k$  = Original nominal state vector of position and velocity at the time  $t_k$ , designed to satisfy some set of target conditions under some set of "integrating" conditions;
- 2)  $\tilde{X}(t_k) = \tilde{X}_k$  = Most recent nominal state vector, only different from  $\bar{X}_k$  after a quasi-linear filtering event that updates the state vector by the estimate;
- 3)  $d\tilde{X}_k$  = "Actual" state vector deviation from most recent nominal trajectory;
- 4)  $\delta\tilde{X}_k$  = Estimated state vector deviation from most recent nominal trajectory;
- 5)  $P_k^+ = E \left[ \left( \delta\tilde{X}_k - d\tilde{X}_k \right) \left( \delta\tilde{X}_k - d\tilde{X}_k \right)^T \right]$  = Statistical description of orbit determination or navigation uncertainty after processing all measurement data up to and including the time  $t_k$ .

The basic cycle of the simulation mode refers to the computational process by which the subscripts on the above quantities are changed from  $k$  to  $k + 1$ . In terms of simulating a real flight, changing the subscripts implies the processing of another measurement. Before proceeding with a step-by-step discussion of this basic cycle, it is worthwhile to point out that, unlike the error analysis mode, the simulation mode is involved in actually processing data to estimate an interplanetary trajectory. Some "actual" trajectory is being flown within the computer and simulated measurements from Earth-based tracking stations are recorded, based upon this "actual" trajectory. These measurements are then processed in a recursive orbit determination algorithm; thus the simulation mode provides a check of the orbit determination procedure's ability to reproduce the "actual" trajectory under a wide set of conditions that might be anticipated on an actual interplanetary mission.

The first step in the basic cycle of the STEAP simulation mode involves calling the measurement scheduling subroutine with the latest trajectory time  $t_k$ . The scheduling subroutine, which has ordered all the measurements to be made during the flight of the simulated trajectory, returns information on the time  $t_{k+1}$  of the next measurement and the kind of measurement (Doppler data from Goldstone, a Canopus-Mars angle, etc) that occurs at that time. Next the event scheduling subroutine is consulted to determine whether or not some computational event (such as guidance) occurs before the next measurement is to be processed. If so, the basic cycle is terminated, control is returned to another part of the program, and the necessary event information is computed.

If no events take place between the time  $t_k$  of the  $k^{\text{th}}$  measurement and the time  $t_{k+1}$  of the  $k + 1^{\text{st}}$  measurement, the basic cycle next uses the virtual mass trajectory mode, available as an n-body subroutine within both the error analysis and simulation modes, for the computation of the original nominal state vector  $\bar{X}_{k+1}$  at the next measurement time. An important point should be stressed here. Within the simulation mode, two separate n-body routines are used, both based on the virtual mass concept. One of these routines, called the actual integrator  $\text{NTM}_1$ , supposedly represents the real world within the program and generates "actual" trajectory data. The second routine, called the assumed integrator  $\text{NTM}_0$ , is used for the original nominal targeted trajectory, the updated nominals, and all estimated trajectories. Its fixed time anomaly arc for trajectory computation may be larger than  $\text{NTM}_1$ , its ephemeris may be approximate or incomplete, and its physical constants used for the n-body package may be assumed in error. The purpose of having two distinct n-body routines available to the simulation mode is perhaps subtle. When actual spacecraft flights take place, the real equations of motion governing the vehicle are never completely known. Thus, to simulate an interplanetary trajectory in a computer program and make it realistic it should be possible to show what happens when the orbit determination algorithm does not have precise knowledge of all the actual quantities involved. The adoption of two different trajectory routines should permit a deeper study of the problems associated with the navigation procedure.

Returning to the basic cycle, after  $NTM_0$  is used to calculate the original nominal state vector  $\bar{X}_{k+1}$  at the new time, the assumed integrator is then used to compute the most recent nominal,  $\tilde{X}_{k+1}$  at the new time. If, prior to  $t_k$ , there have been no quasi-linear filtering events that have updated the nominal, then the most recent nominal state vector  $\tilde{X}_{k+1}$  is the same as the original  $\bar{X}_{k+1}$ .

The next steps in the basic cycle for the simulation mode are the same as those in the error analysis mode. The new navigation uncertainty covariance matrix  $P_{k+1}^+$ , the Kalman gain matrix  $K_{k+1}$ , and the residual uncertainty matrix  $S_{k+1}$  are computed from the basic Kalman recursive estimation algorithm,

$$\begin{aligned} P_{k+1}^- &= \Phi_{k+1,k} P_k^+ \Phi_{k+1,k}^T + Q_{k+1,k} \\ S_{k+1} &= H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1} \\ K_{k+1} &= P_{k+1}^- H_{k+1}^T S_{k+1}^{-1} \\ P_{k+1}^+ &= P_{k+1}^- - K_{k+1} H_{k+1} P_{k+1}^- \end{aligned} \tag{133}$$

where, as before,  $\Phi_{k+1,k}$  is the required state transition matrix and  $Q_{k+1,k}$  is the process noise matrix. The residual uncertainty matrix  $S_{k+1}$  describes the statistics of the actual measurement residual and is very important in what is known as adaptive filtering, a topic to be discussed in Section D of this chapter. An extension of the current program to permit the use of adaptive filtering would not be difficult.

The next steps of the STEAP simulation mode basic cycle are concerned with the generation of the "actual" state vector deviations  $d\tilde{X}_{k+1}$  at the time  $t_{k+1}$ , referenced to the most recent nominal trajectory. The complete "actual" state vector at  $t_k$ , namely  $X_k + d\tilde{X}_k$ , is first used in the actual integrator  $NTM_1$

to obtain a quantity  $Z_{k+1}$ , which would be the actual state vector at  $t_{k+1}$  if there were no unmodeled accelerations between  $t_k$  and  $t_{k+1}$ . However, the program allows for the addition of unmodeled accelerations  $\ddot{\delta X}$ ,  $\ddot{\delta Y}$ , and  $\ddot{\delta Z}$  that integrate into a state vector addition  $\omega_{k+1}$ . Thus, the required quantity  $d\tilde{X}_{k+1}$  becomes

$$d\tilde{X}_{k+1} = Z_{k+1} + \omega_{k+1} - \tilde{X}_{k+1} \quad (134)$$

where  $\omega_{k+1}$  is the actual unmodeled addition to the state vector over the defined interval.

Unmodeled accelerations are permitted to corrupt the actual state vector for a very definite purpose. Along an interplanetary flight, many possible sources of mechanical difficulty onboard the vehicle could give rise to small accelerations. It is important, for the purposes of the simulation, to determine how the orbit determination algorithm reacts in the presence of small accelerations about which the algorithm itself has no specific knowledge.

Another digression concerning the underlying philosophy of the simulation mode is now warranted. Recall that its purpose is essentially to test a specific navigation and guidance process, insofar as is possible, under real conditions. Four key assumed statistical descriptions are used by the estimation algorithm to produce the optimal estimate of the state vector. These four are the injection covariance  $P_0$ , the process noise matrices  $Q_{k+1,k}$ , the measurement noise matrices  $R_k$ , and the midcourse correction execution error covariances  $Q_j$ . All of the matrices represent assumed errors and their probabilistic descriptions. Obviously the convergence of the estimated trajectory to the actual trajectory, for a real flight, is a function of the accuracy of these a priori statistics. To test the orbit determination and guidance process within the simulation mode, actual injection errors  $\Delta X_0$ , actual midcourse execution errors, actual unmodeled accelerations, and actual measurement noise statistics  $R_k$  may be specified by the user. These specifications permit the study of the effect of bad a priori statistics on the success of the defined navigation and guidance algorithms.

The final steps of the simulation mode basic cycle are concerned with the computation of the last of the five needed quantities, the estimated deviation  $\delta\tilde{X}_{k+1}$  from the most recent



nominal state vector. First, the most recent nominal state vector  $\tilde{X}_{k+1}$  is used to determine the most recent nominal observation  $\tilde{Y}_{k+1}$  at the time  $t_{k+1}$ . From the measurement scheduling subroutine, the kind of measurement made at  $t_{k+1}$  is known. For whatever measurement kind is used, there is a functional relationship relating the measurement to the state vector that is given by

$$\tilde{Y}_{k+1} = f(\tilde{X}_{k+1}, p)$$

where  $p$  are parameters, such as station location coordinates, that are used in the computation of  $\tilde{Y}_{k+1}$ . Next the "actual" state vector at  $t_{k+1}$ , given by  $\tilde{X}_{k+1} + d\tilde{X}_{k+1}$ , is used together with actual station location coordinates  $p$  to compute  $\tilde{Y}_{k+1}$ , the "actual" measurement that would have been made in the absence of any instrumentation errors, according to

$$\tilde{Y}_{k+1} = f(\tilde{X}_{k+1}, t_{k+1}, \bar{p})$$

To this quantity is added the "actual" measurement bias  $b$ , as well as the sampled measurement noise  $v_{k+1}$ , for this particular measurement, obtained by randomly sampling the actual measurement noise covariance matrix  $R_{k+1}$  (which may or may not equal the assumed measurement noise  $R_{k+1}$ ). Thus the "actual" measurement at the time  $t_{k+1}$ , which is the only output of the actual portion of the simulation mode that is available to the estimation algorithm, is given by  $Y_{k+1}^a$ , where

$$Y_{k+1}^a = \tilde{Y}_{k+1} + v_{k+1} + b$$

The measurement that the estimation algorithm expects or predicts is next calculated as

$$Y_{k+1}^e = \tilde{Y}_{k+1} + H_{k+1} \phi_{k+1,k} \delta \tilde{X}_k$$

where  $H_{k+1}$  is the observation matrix relating linear perturbations in the measurements at the time  $t_{k+1}$  to linear perturbations about the most recent nominal state vector at the same time.

If the estimated trajectory and the actual trajectory were coincident at the time  $t_k$  and the linearity assumptions regarding the computation of  $H_{k+1}$  and  $\phi_{k+1,k}$  were completely valid, and if, in addition, the measurement instrumentation were perfect, then the expected and actual measurements would agree. The measurement residual  $\epsilon_{k+1}$  at the time  $t_{k+1}$  is next calculated as

$$\epsilon_{k+1} = y_{k+1}^a - y_{k+1}^e$$

Finally, using the standard Kalman filtering equations, the estimated deviation  $\delta \tilde{X}_{k+1}$  from the most recent nominal state vector  $\tilde{X}_{k+1}$  at the time  $t_{k+1}$  is given by

$$\delta \tilde{X}_{k+1} = \phi_{k+1,k} \delta \tilde{X}_k + K_{k+1} \epsilon_{k+1}$$

where  $K_{k+1}$  is the Kalman gain matrix defined earlier.

The presence of large residuals,  $\epsilon_{k+1}$  indicates one of two things: either the noise on the measurements is very high, or the estimated trajectory is not tracking the actual trajectory. The concept of adaptive filtering mentioned in a preceding paragraph involves checking the magnitude of the residual vector  $\epsilon_{k+1}$  against its assumed a priori statistics  $S_{k+1}$  as computed by the estimation algorithm. A statistical inconsistency between the two is an indication of filter divergence.

The basic cycle of the simulation mode is essentially complete when the five needed quantities are determined. However, within the simulation mode of STEAP, three other vectors of particular interest at a guidance event are computed. These are the actual orbit determination uncertainty and both the actual and estimated deviations from the original nominal.

## B. Guidance Event

Just as in the error analysis mode, the most complex computational event in the simulation mode is the guidance event. In the simulation mode guidance event algorithm, not only is the execution error covariance matrix computed, but also a commanded and

an actual correction must be calculated, based on some guidance policy and model of the execution process.

Let  $t_j$  be the time of the  $j^{\text{th}}$  guidance correction and let  $t_k$  be the time of the last measurement before the correction. From the basic cycle of the simulation mode, the five quantities  $\bar{X}_k$ ,  $\tilde{X}_k$ ,  $d\tilde{X}_k$ ,  $\delta\hat{X}_k$ , and  $P_k^+$ , defined in the preceding section, are available at the time  $t_k$ . They are propagated forward to the time  $t_j$  in a normal fashion that mirrors the basic cycle in the absence of measurements. The assumed n-body routine  $NTM_0$  is used to determine  $\bar{X}_j$  and  $\tilde{X}_j$  from  $\bar{X}_k$  and  $\tilde{X}_k$ . The actual unmodeled state vector addition  $\omega_j$  is calculated from the unmodeled acceleration specified over the interval  $\Delta t = t_j - t_k$  and, after computing  $Z_j$  from  $\tilde{X}_k + d\tilde{X}_k$  by using  $NTM_1$  as in the basic cycle,  $d\tilde{X}_j$  becomes

$$d\tilde{X}_j = Z_j + \omega_j - \tilde{X}_j \quad (135)$$

Finally, the remaining two values  $\delta\hat{X}_j$  and  $P_j^-$  follow the prediction equations of a Kalman estimator,

$$\delta\hat{X}_j = \Phi_{j,k} \delta\hat{X}_k \quad (136)$$

$$P_j^- = \Phi_{j,k} P_k^+ \Phi_{j,k}^t + Q_{j,k}$$

where, as always,  $\Phi_{j,k}$  is the state transition matrix and  $Q_{j,k}$  is the process noise matrix.

At the guidance event, only three of the quantities are additionally changed before returning to the basic cycle. Both  $\bar{X}_j$  and  $\tilde{X}_j$ , the original and most recent nominal state vectors respectively, are not altered by the remainder of the guidance event algorithm. However, the other three are operated on at the guidance event according to

$$\begin{aligned}
P_j^+ &= P_j^- + \begin{bmatrix} 0 & 0 \\ 0 & \tilde{Q}_j \end{bmatrix} \\
\delta \tilde{X}_j^+ &= \delta \tilde{X}_j^- + \begin{bmatrix} 0 \\ \Delta V_j \end{bmatrix} \\
d\tilde{X}_j^+ &= d\tilde{X}_j^- + \begin{bmatrix} 0 \\ \Delta V_j + \delta \Delta V_j \end{bmatrix}
\end{aligned} \tag{137}$$

where  $\tilde{Q}_j$  is the 3 x 3 covariance matrix of execution errors,  $\Delta V_j$  is a 3 x 1 vector of commanded correction, and  $\delta \Delta V_j$  is the "actual" error in the correction. All three of these quantities, as well as other auxiliary values, are computed within the guidance event algorithm of the STEAP simulation mode.

Three guidance policies are also permitted in the simulation mode. They are the same three -- FTA, three-variable B-plane, and two-variable B-plane -- as in the error analysis mode. For whatever guidance policy is specified a guidance matrix  $\Gamma_j$  is first calculated according to the same equations given earlier in Chapter V, Section E. As part of the derivation of  $\Gamma_j$ , the variation matrix  $\eta_j$ , of dimensionality either 3 x 6 or 2 x 6 according to the policy, is computed that satisfies

$$\delta t_c = \eta_j \delta X_j$$

where  $\delta t_c$  are linear variations in the target conditions and  $\delta X_j$  are linear variations in the state vector at the time of the  $j^{\text{th}}$  guidance correction. Within the simulation mode,  $\eta_j$  is based on deviations around the most recent nominal to ensure greater validity of the linearizing approximation.

Once  $\Gamma_j$  has been computed, the estimated deviation  $\delta \hat{X}_j$  from the original nominal is determined from

$$\hat{\delta X}_j = \tilde{X}_j + \hat{\delta \tilde{X}}_j - \bar{X}_j$$

Recall that the original nominal trajectory was assumed to satisfy certain specified target conditions and thus the midcourse correction  $\Delta V_j$  is chosen to null target condition deviations from the original nominal. Based on a linear, impulsive guidance scheme, the commanded correction  $\Delta V_j$  is simply

$$\Delta V_j = \Gamma_j \hat{\delta X}_j \quad (138)$$

For comparison purposes and to show the sensitivity of  $\Delta V_j$  to navigation errors at the time of the  $j^{\text{th}}$  correction, an auxiliary vector  $\underline{\Delta V}_j$ , called the perfect correction, is determined from the actual state vector deviation  $dX_j$  from the original nominal. Since  $dX_j = \tilde{X}_j + d\tilde{X}_j - \bar{X}_j$ , the perfect correction is given by

$$\Delta V_j = \Gamma_j dX_j \quad (139)$$

The vector  $\Delta V_e$ , called the error in the correction due to navigation uncertainty, may be written as

$$\Delta V_e = \underline{\Delta V}_j - \Delta V_j \quad (140)$$

The commanded correction  $\Delta V_j$  is now used to determine the execution error covariance matrix  $\tilde{Q}_j$ . Recall that in the guidance event algorithm for the error analysis mode, the execution error matrix  $\tilde{Q}_j$ , in the absence of estimation, was computed from an effective midcourse correction vector " $E[\Delta V_j]$ ". The calculation of the execution error matrix  $\tilde{Q}_j$  in the simulation mode, as would be expected, is exactly the same, except that now the components of  $\Delta V_j$ , the commanded correction, are used to determine the elements of the matrix  $\tilde{Q}_j$ .

The probabilistic uncertainties in the target conditions before and after the correction, defined by the matrices  $W_j^-$  and  $W_j^+$  and discussed in Chapter V, Section E, are next calculated to show the target condition gains, or likely gains, from making the correction.

The only quantity remaining for the guidance event algorithm to compute is the "actual" execution error  $\delta\Delta V_j$ . The same execution error model used in the error analysis mode is again assumed. If there are a total of  $n$  midcourse corrections, then a  $4 \times n$  array of actual execution errors are input to the program. For the  $j^{\text{th}}$  midcourse correction, these actual errors, which may or may not be consistent with the input variances on the same quantities used for the calculation of  $\tilde{Q}_j$ , are

$s^j$  = actual resolution error at  $j^{\text{th}}$  correction;

$k^j$  = actual proportionality error at  $j^{\text{th}}$  correction;

$d\alpha^j, d\beta^j$  = actual angular pointing errors at  $j^{\text{th}}$  correction.

Following the derivation of the execution error  $\delta\Delta V$  given in Chapter V, Section E, the actual execution error at the time of the  $j^{\text{th}}$  correction is found to be

$$\begin{aligned} \delta\Delta V_j = & \left[ \Delta V_X \left( s^j + \frac{k^j}{\rho} \right) + \frac{\rho \Delta V_Y \delta\alpha^j + \Delta V_X \Delta V_Z \delta\beta^j}{u} \right] \hat{i} \\ & + \left[ \Delta V_Y \left( s^j + \frac{k^j}{\rho} \right) + \frac{\Delta V_Y \Delta V_Z \delta\beta^j - \rho \Delta V_X \delta\alpha^j}{u} \right] \hat{j} \\ & + \left[ \Delta V_Z \left( s^j + \frac{k^j}{\rho} \right) - u \delta\beta^j \right] \hat{k} \end{aligned} \quad (141)$$

where

$$u = \left( \Delta V_X^2 + \Delta V_Y^2 \right)^{1/2}$$

$$\rho = \left( \Delta V_X^2 + \Delta V_Y^2 + \Delta V_Z^2 \right)^{1/2}$$

$\hat{i}, \hat{j}, \hat{k}$  are the unit vectors in XYZ heliocentric elliptic system and  $\Delta V_X, \Delta V_Y, \Delta V_Z$  are the heliocentric elliptic components of the commanded correction  $\Delta V_j$ .

Before returning to the basic cycle with the necessary computed information, several additional vectors of interest are calculated. The two most important are the error at the target due to navigation uncertainty and the error at the target due to the execution error. Letting  $\eta_j$  again be the variation matrix, the error at target due to navigation uncertainty, defined by  $\epsilon_n$ , is give by

$$\epsilon_n = \eta_j \begin{bmatrix} \tilde{dX}_j \\ -\tilde{\delta X}_j \end{bmatrix} \quad (142)$$

Similarly, the error at the target due to the execution error may be calculated as

$$\epsilon_{ex} = \eta_j \begin{bmatrix} 0 \\ \delta \Delta V \end{bmatrix} \quad (143)$$

The "actual" error at the target after the correction is the vector sum of the two above quantities.

### C. Other Events

Three other events are permissible when STEAP is operating in a simulation mode. The first two, eigenvector events and prediction events, are just the same as they are in the error analysis mode except that all five quantities must first be propagated forward to the event time. This is accomplished in the same fashion as it is at the beginning of a guidance event and warrants no additional comment.

The third event is called a quasi-linear filtering event. At a quasi-linear filtering event the original nominal trajectory is updated by using the most recent estimate. The purpose of the update is to combat divergence due to the possible invalidity of the linearity assumption that is the basis for the estimation algorithm being used. Specifically, updating the nominal trajectory results in better computations of the  $\phi$  and  $H$  matrices being used in the weighting at the time of processing each measurement.

Let  $t_E$  be the time of a quasi-linear filtering event and assume that  $\tilde{X}_E$ ,  $\tilde{X}_E$ ,  $\tilde{\delta X}_E$ ,  $\tilde{dX}_E$ , and  $P_E$  have all resulted from the usual propagation of the five quantities forward from

last measurement. Defining by  $\bar{X}_E^+$ ,  $\tilde{X}_E^+$ , etc, the new values after the quasi-linear filtering event, the computations are

$$\begin{aligned}\bar{X}_E^+ &= \bar{X}_E^- \\ \tilde{X}_E^+ &= \tilde{X}_E^- + \delta\tilde{X}_E^- \\ d\tilde{X}_E^+ &= d\tilde{X}_E^- - \delta\hat{X}_E^- \\ \delta\hat{X}_E^+ &= 0 \\ P_E^+ &= P_E^-\end{aligned}\tag{144}$$

From the equations above it is easy to see that the quasi-linear filtering event effectively just updates the most recent nominal by adding to it the estimate of deviations from the previous most recent nominal.

Due to internal complexities of the program, when augmented states are being treated in the simulation mode, only the position and velocity components of the state vector are updated at a quasi-linear filtering event. Since all the additional parameters in the augmented state are constants that are reasonably well-known initially, this limitation does not appear to be significant.

#### D. Divergence and Other Problems

One of the purposes of creating such a detailed and extensive simulation mode was to study the problem of filter divergence. The problem of divergence in a recursive navigation process and a companion difficulty, computational nonobservability, are the subjects of this section.

Strictly speaking, when divergence occurs in a navigation process, the navigation is failing to navigate properly. The phenomenon of divergence never appears in an error analysis



mode because no actual estimation is taking place and only covariance matrices are being propagated. In a computer simulation such as the STEAP simulation mode, where an "actual" trajectory is being flown and concurrently estimated by a navigation algorithm, filter divergence refers to the failure of the estimated trajectory to converge, within reasonable bounds specified by the covariance matrices, to the "actual" simulated trajectory. For real-world orbit applications, where the actual trajectory is never known, divergence is occurring when the residual difference between predicted and actual observation vectors becomes increasingly large.

In either computer simulation or a real orbit determination procedure, divergence in the recursive filter manifests itself as a statistical inconsistency between the measurement residuals and the filtering algorithm. Recall that at each step of the recursive process, the matrix  $S_{k+1} = H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1}$  is computed. This matrix defines the a priori statistics associated with the measurement residual  $\epsilon_{k+1}$ . The measurement residual  $\epsilon_{k+1}$  should represent a sample from the population defined by  $S_{k+1}$ . When divergence occurs, a group of successive residuals appear less and less likely, statistically, to have been sampled from their covariances  $S_{k+1}$ .

To illustrate the divergence manifestation in terms of measurement residuals inconsistent with their a priori covariances, assume that a scalar range-rate measurement is being taken along an interplanetary orbit. For scalar measurements, the matrix  $S_{k+1}$  is a scalar residual variance, call it  $\sigma_\epsilon^2$ . Suppose that for the first two hundred measurements, each measurement residual was compared to its statistical variance by solving

$$\epsilon_j = K_j \sigma_{\epsilon_j}, \quad j = 1, 2, \dots, 100 \quad (145)$$

for the value  $K_j$ . Suppose further that a frequency histogram of the values  $K$  produced

<u>Internal</u>	<u>No.</u>	<u>Percent</u>	<u>Theoretical Percent</u>
$-\frac{1}{2} \leq K < \frac{1}{2}$	79	39.5	38.3
$-\frac{3}{2} \leq K < -\frac{1}{2}$	55	27.5	24.17
$-\frac{5}{2} \leq K < -\frac{3}{2}$	9	4.5	6.06
$K < -\frac{5}{2}$	4	2.0	0.62
$-\frac{1}{2} \leq K < \frac{3}{2}$	38	19.0	24.17
$\frac{3}{2} \leq K < \frac{5}{2}$	13	6.5	6.06
$K < \frac{5}{2}$	2	1.0	0.62

Without subjecting the above data to a rigorous chi-square test of hypothesis, it should be clear that the measurement residuals, each of which is assumed to be an uncorrelated, Gaussian, mean zero random variable, are more or less consistent with their statistics; that is, the ensemble values for the measurement residuals look reasonable in terms of their a priori variances used within the estimation algorithm.

Now suppose that the next nine values of  $K_j$ , determined in the same fashion from the nine measurements following the two hundredth, are give by the sequence

$$\begin{array}{lll}
 K_{201} = -2.4 & K_{204} = -4.7 & K_{207} = -6.2 \\
 K_{202} = -3.7 & K_{205} = -5.1 & K_{208} = -7.1 \\
 K_{203} = -4.4 & K_{206} = -5.5 & K_{209} = -7.8
 \end{array} \tag{146}$$

From the underlying assumptions of the navigation process, each of these events, taken singly, is extremely unlikely. However, the sequence of values given is almost totally unlikely and

should represent a dead giveaway that divergence is occurring. Without pursuing the mathematics too far, it should be stressed that if the nine values given in equation (146) were supposedly chosen at random from a normal distribution with near zero and unit variance, the governing distribution would fail every test of statistical hypothesis. Such values for  $K$  indicate that something in the estimation process is definitely wrong: the most likely candidate for the error is the assumed a priori  $S$  matrix used for weighting by the estimation algorithm.

The hypothetical example given above is typical of the divergence phenomenon that recurs in complex orbit determination processes. Often, the process converges initially and then, after many measurements have been taken, divergence begins. A general explanation for this is that the covariance matrices associated with the estimated state vector become overly optimistic and, subsequently, tend to disregard the new measurement data in the weighting process.

The general cause of divergence is modeling insufficiency. For most real problems, everything about the dynamical system and the observations being treated by the filter is not known exactly. Unless the estimation algorithm acknowledges, in some fashion, the incomplete understanding of the governing equations, divergency may result. A familiar source of insufficient modeling is the dynamic equations themselves. All the forces acting on an interplanetary spacecraft are never known exactly. In addition, the filtering algorithm speaks on perturbation equations resulting from a linearization about some reference dynamic state. Thus the procedure is working with appropriate equations and unless process noise is added to the computational algorithm, the Kalman filter "thinks" it knows the exact equations of motion, whereas in reality it does not.

Divergence can also result from other model inadequacies. Among the most frequent causes are failure to account for measurement nonlinearities when the measurements themselves are very accurate, neglect of correlated errors between sequences of measurements taken by the same instruments, and overly optimistic a priori error statistics describing the measurement noise. Within the simulation mode of STEAP, the effect of all these model inadequacies on a specific reference trajectory can be tested.

Many possible solutions to the problem of divergency have been postulated and investigated. Two of the methods of divergence prevention have been included in STEAP and, because they are the simplest, they will be discussed first.

In an earlier section of this report the modeling of a process noise matrix  $Q_{k+1,k}$  was discussed. Between measurements in the estimation algorithm of STEAP, the state vector associated covariance matrix is propagated according to the equation

$$P_{k+1}^- = \Phi_{k+1,k} P_k^+ \Phi_{k+1,k}^T + Q_{k+1,k} \quad (147)$$

This  $Q$  matrix, the size of which is specified by input to the program, increases the magnitude of the key diagonal elements in the covariance matrix. Because divergence generally occurs when the state vector associated covariances become unduly optimistic and additional measurements are weighted very slightly, the addition of  $Q$  represents an attempt to systematically downgrade the dynamics information in favor of the measurements.

Although the addition of a proper  $Q$  matrix will impede divergence, unless its size is determined by physical considerations it can also slow convergence. Most often the  $Q$  matrix is somewhat arbitrary and its exact influence on the estimation algorithm is not clearly understood. Thus attempts should be made, based on the modeling for a particular problem, to ensure that the elements of  $Q$  are realistic.

A second method of divergence prevention included in STEAP involves what is known as quasi-linear filtering. The fundamental estimation process assumes that variations about the nominal trajectory and nominal measurements are linear. In the case of highly accurate measurements, which is usually the case of interplanetary spacecraft tracked by the DSIF, measurement nonlinearities become significant model inadequacies when the actual trajectory is only slightly different from the nominal. Quasi-linear filtering essentially permits more accurate computation of the linear perturbation matrices  $\Phi$  and  $H$ . This is accomplished by updating the original nominal trajectory, based on the estimated state vectors coming from the navigation algorithm, and then computing both the state transition and observation matrices in terms of linear perturbations about the updated nominal.

The inclusion of augmented state options in STEAP, which results in larger covariance matrices associated with the state vector because of uncertainties in dynamic or measurement constants that are treated by the process, is an indirect attempt to combat divergence in the presence of known modeling insufficiencies. However, within the current version of STEAP, the augmented state vector options operate only in what is known as

a "solve for" mode; that is, the error analysis mode systematically reduces the uncertainties in the augmented parameters as if they were being estimated by the underlying algorithm. Similarly, the simulation mode actually estimates the augmented quantities in the state vector.

Treatment of errors in dynamic or measurement parameters without actually estimating them or augmenting the state is generally called the "consider" mode. Essentially the "consider" divergence prevention method actually calculates the process noise matrix  $Q$  over each interval based on physical assumptions on the size of the parameter uncertainties. It is hoped that such a practical approach to the computation of  $Q$  will be included in STEAP in the near future.

Of the other methods for handling filter divergence that have been suggested in the literature, the strongest appears to be adaptive filtering. Reference 13 explains the theoretical basis for several kinds of adaptive filtering schemes. The essential idea of adaptive filtering is the feedback of actual measurement residuals into the covariance matrix propagation process. Earlier it was pointed out that a sign of filter divergence is a statistical inconsistency between the measurement residuals  $\epsilon_{k+1}$  and their assumed a priori covariance matrices  $S_{k+1}$  used by the estimation algorithm. In adaptive filtering, this statistical inconsistency is used to change the assumed a priori statistics, on both the dynamics and the measurements until the residuals and their updated covariances are more or less consistent. Optimal implementation of adaptive filtering is being pursued by several researchers in the field.

Another problem associated with interplanetary orbit determination that can be studied with the STEAP simulation mode is that of computational nonobservability. Because this problem threatens to occur whenever strictly Earth-based tracking is being used to determine the orbit of a spacecraft around the Moon or another planet, it warrants attention.

In classical batch-processing algorithms, observability does not exist when a key matrix inverse used to determine the estimate does not exist. In a recursive algorithm, nonobservability manifests itself when one of the correlation coefficients relating uncertainties in different elements of the state vector has unit magnitude. Physically this means that the navigation process cannot observe or estimate the two quantities that are either positively or negatively correlated uniquely. The orbit determination procedure has no unique convergence in this case.

When the correlation coefficients relating uncertainties in two elements of the state vector are very close to unity in magnitude, then the underlying estimation algorithm is very unstable. Although theoretically a unique solution still exists, any model inadequacies can produce wild gyrations in the estimated solutions. Preliminary studies with STEAP of orbit determination processes using Earth-based tracking for spacecraft in Moon or Mars orbits indicate that the above orbit determination instability, called computational nonobservability, is very much a real problem.

## VII. NUMERICAL RESULTS

This chapter deals exclusively with results obtained from using the Simulated Trajectories Error Analysis Program. The commentary in this chapter should enhance the user's knowledge and appreciation of the studies that can be conducted using the program.

There are four separate sections in this chapter. Each section is concerned with example runs obtained from one particular operational mode of the program. The first section presents example results from the trajectory mode. Sample runs with the targeting mode are contained in the second section. The lengthy third and fourth sections discuss example runs made with the many options of the error analysis and simulation modes.

### A. Trajectory Mode

In this section of the report, a detailed description of the uses of the trajectory mode will be presented. It is assumed that the analytic techniques used within the program -- the virtual mass integration scheme, the ephemeris employing mean conic section orbital elements, etc -- are now well known. Hence the detailed discussion to follow will concentrate on the uses of the trajectory mode and will attempt, insofar as is possible, to explain how the program may be used most efficiently. Many sample runs have been generated and it is hoped that the knowledge gleaned from these runs can be successfully passed on.

One of the most important variables to be understood when activating the trajectory mode is what is called the accuracy level. As mentioned in an earlier section, the accuracy level is a variable developed when the virtual mass program was used for analysis of Earth-Moon trajectories. At that time it referred to the likely ratio of the error in pericynthion passage distance to the distance itself. Because of the inner logic of the program its use has been retained in the program, although it is now, strictly speaking, a dummy variable with no intrinsic meaning when applied to interplanetary trajectories.

For any set accuracy level, the program automatically computes the step size to be used in the "integration" (the word "integration" is placed in quotes because, as is known, the virtual mass program does not ever use any quadratures). The computational

step size in the program is a fixed true anomaly arc relative to the virtual mass. As can be seen from table 1, an accuracy level of  $1.25 \times 10^{-4}$  corresponds to a true anomaly increment of 35.946 mrad. Similarly, if the external accuracy level is set at  $2 \times 10^{-7}$ , then each computational step within the program allows a time passage that is commensurate with a true anomaly arc of 1.698 mrad on a conic section arc about the most recent virtual mass. By determining the step size from a fixed true anomaly arc relative to the effective force center, the "integration" automatically takes smaller steps when the spacecraft is in a critical region near a planet. Conversely, large steps are automatically taken during the essentially heliocentric phase of an interplanetary orbit.

The first detailed study on the capabilities of the trajectory mode was aimed at determining the effect of varying the accuracy level (and hence the step size used in the "integration") on the resulting computed trajectory. The results of this study are summarized in tables 1 and 2. Two sample trajectories were chosen, one passing very close to Mars and chosen in the middle of the tentative Viking launch window, and another direct shot from the Earth to Jupiter passing within four Jupiter radii of the large planet. As would be expected, the Mars-Viking run is extremely sensitive to any changes in the computational procedure and the results of increasing the accuracy level for that base trajectory do not demonstrate the relatively "clean convergence" shown by table 2 for the Jupiter trajectory.

Examining table 1 in some detail gives a fairly definite picture of the effect of the accuracy level on the computations. Included in the table are the results of five separate computer runs. For each of the runs, everything was exactly the same (that is, same injection conditions, same ephemeris computation, same printout specifications, etc) except for the accuracy level employed. Table 1 lists, for each run, the total CDC 6500 computer time for the run, the number of individual computational steps within the run, the computational time for each increment within each of the runs, the number of steps to the Martian sphere of influence, the calculated trajectory time to both the Martian sphere of influence and the point of Martian closest approach, the radius and velocity at closest approach, and the quantities  $V_{HE}$ ,  $B \cdot T$ ,  $B \cdot R$  (defined in the usual manner) at the Martian sphere of influence. It should be mentioned that runs at a higher accuracy level than  $2 \times 10^{-7}$  did not produce significant differences from the values listed for Run 5404; thus it can be assumed,



TABLE 1.- MARS-VIKING RUN, SUMMARY OF ACCURACY ANALYSIS

Run no.	Accuracy level	True anomaly increment, mrad	Computer time, sec	Increment number	Time per increment, sec	Increments to SOI	Flight time to SOI, days	Flight time to CA, days	R <sub>CA</sub> , km	V <sub>CA</sub> , km/sec	V <sub>HE</sub> , km/sec	B-T, km	B-R, km
5400	1.25 x 10 <sup>-4</sup>	35.946	24.864	800	0.0322	526	195.57361	197.82431	51 412	3.126	2.8722	-34363	44 558
5401	2.5 x 10 <sup>-5</sup>	16.570	48.876	2 079	0.0235	1073	195.27461	197.48881	4 820	5.097	2.8807	-243	8 460
5402	5 x 10 <sup>-6</sup>	7.696	93.011	4 477	0.0208	2218	195.21241	197.4219	3 805	5.543	2.8825	7521	550
5403	1 x 10 <sup>-6</sup>	3.601	182.423	9 270	0.0197	4639	195.1992	197.4125	5 234	4.956	2.8829	9231	-983
5404	2 x 10 <sup>-7</sup>	1.698	373.107	19 503	0.0196	9738	195.2012	197.4150	5 454	4.889	2.8828	9487	-725

Note: Ephemeris = Earth, Mars, Sun, Jupiter.

TABLE 2.- JUPITER RUN, SUMMARY OF ACCURACY ANALYSIS

Run no.	Accuracy level	Computer time, sec	Increment Number	Time per increment, sec	Increments to SOI	Flight time to SOI, days	Flight time to CA, days	R <sub>CA</sub> , km	V <sub>CA</sub> , km/sec	V <sub>HE</sub> , km/sec	B-T, km	B-R, km
6200	1.25 x 10 <sup>-4</sup>	19.584	838	0.0234	481	646.3505	715.9043	279 853	30.864	7.2284	695 121	677 427
6201	2.5 x 10 <sup>-5</sup>	33.102	1 738	0.0190	969	646.1948	715.7153	284 528	30.623	7.2319	698 130	695 760
6202	5 x 10 <sup>-6</sup>	62.196	3 669	0.0169	2023	646.1570	715.6703	285 883	30.554	7.2327	699 615	700 040
6203	1 x 10 <sup>-6</sup>	123.993	7 752	0.0160	4242	646.1363	715.6588	286 284	30.533	7.2329	700 185	701 105
6204	2 x 10 <sup>-7</sup>	253.726	16 348	0.0155	8910	646.1316	715.6536	286 504	30.522	7.2330	700 460	701 598

Note: Ephemeris = Earth, Sun Jupiter.

for this base trajectory anyway, that an accuracy of  $2 \times 10^{-7}$  is sufficient for generating a very accurate trajectory computation (no equinox or ecliptic rotations, no oblateness for the Earth, no solar pressure, etc), there will probably be very few times when such a high accuracy level is needed.

The key factor governing total computation time is the number of increments used to generate the trajectory. Basically, the total time used by the program to calculate the trajectory is seen to be between 0.02 and 0.03 sec per step used in the calculation. Although the discrepancies in terms of target conditions for the various accuracy levels are quite high when viewed from a totally precise n-body trajectory computation, it should be remembered that the trajectory itself is tremendously sensitive to any changes, a fact that is demonstrated by the detailed figures that follow.

Comparing two different trajectories on which the accuracy study was made (tables 1 and 2), a considerable amount of agreement is found in essential quantities. For example, an accuracy level of  $1.25 \times 10^{-4}$  produces a B.T and B.R bias of roughly 45 000 km when compared to the highly accurate ( $2 \times 10^{-7}$ ) run for the Martian trajectory. The Jovian trajectory, admittedly less sensitive, shows a B.T, B.R change of less than 25 000 km when the accuracy level is increased from  $1.25 \times 10^{-4}$  to  $2 \times 10^{-7}$ . Similarly, increasing the accuracy level from  $5 \times 10^{-6}$  to  $2 \times 10^{-7}$  makes a B.T, B.R change of about 1500 km in both the base Martian and the base Jovian trajectories.

Although the target planet changes may seem alarmingly large, examination of tables 3 thru 5 and figures 6 and 7 show that it is the high sensitivities of these trajectories that are accounting for the differences. For example, consider table 3. That table gives the aerocentric ecliptic deviations of each of the accuracy level runs from the most accurate run ( $2 \times 10^{-7}$ ), which is Run 5404. Six days into the trajectory, which is well past the Earth sphere of influence, the least accurate trajectory considered (accuracy level  $1.25 \times 10^{-4}$ ) only varies from the most accurate trajectory by 500 km and 1.8 m/sec. Because of the sensitivity of the trajectory, these differences are magnified to 109 000 km and 15.366 m/sec at 194 days, which is near the Martian sphere of influence. Similarly, the next most accurate trajectory (Run

TABLE 3.- MARS-VIKING RUN, DETAILED ACCURACY ANALYSIS  
FOR 6, 30, AND 60 DAYS

	5400	5401	5402	5403
6 days				
$\Delta y$ , km	-408	-69	1	11
$\Delta y$	350	84	25	10
$\Delta z$	-36	-5	1	2
$\Delta R$	<u>538.8</u>	<u>108.8</u>	<u>25.04</u>	<u>15</u>
$\Delta \dot{x}$ , m/sec	-1.339	-.251	-.022	+.017
$\Delta \dot{y}$	+1.199	+.277	+.074	+.025
$\Delta \dot{z}$	-0.093	-.015	.001	.003
$\Delta v$	<u>1.80</u>	<u>.374</u>	<u>.007</u>	<u>.030</u>
30 days				
$\Delta x$ , km	-8 360	-1 771	-302	1
$\Delta y$	6 660	1 530	366	94
$\Delta z$	-328	-57	0	9
$\Delta R$	<u>10 694</u>	<u>2 341</u>	<u>475</u>	<u>94.4</u>
$\Delta \dot{x}$ , m/sec	-3.681	-.781	-.136	-.002
$\Delta \dot{y}$	2.765	.628	.147	.035
$\Delta \dot{z}$	-.144	-.026	-.001	.003
$\Delta v$	<u>4.606</u>	<u>1.003</u>	<u>.200</u>	<u>.035</u>
60 days				
$\Delta x$ , km	-21 200	-4 520	-786	15
$\Delta y$	13 500	3 110	726	174
$\Delta z$	-778	-143	-7	15
$\Delta R$	<u>25 200</u>	<u>5 488</u>	<u>1 070</u>	<u>175.3</u>
$\Delta \dot{x}$ , m/sec	-6.056	-1.288	-.266	-.006
$\Delta \dot{y}$	2.208	.498	.113	.025
$\Delta \dot{z}$	-.201	-.039	-.004	.002
$\Delta v$	<u>6.449</u>	<u>1.382</u>	<u>.253</u>	<u>.026</u>
<p><u>Note:</u> 1. All <math>\Delta</math> values are referenced to components of Run 5404 at that time.</p> <p>2. Coordinates are areocentric ecliptic.</p>				

TABLE 4.- MARS-VIKING RUN, DETAILED ACCURACY ANALYSIS  
FOR 90, 120, AND 150 DAYS

	5400	5401	5402	5403
90 days				
$\Delta x$ , km	-40 100	-8 500	-1470	-21
$\Delta y$	16 500	3 780	898	225
$\Delta z$	-1.410	-272	-25	19
$\Delta R$	<u>43 380</u>	<u>9 307</u>	<u>1723</u>	<u>227</u>
$\Delta \dot{x}$ , m/sec	-8.044	-1.696	-.284	.004
$\Delta \dot{y}$	-.236	-.030	.012	.016
$\Delta \dot{z}$	-.282	-.058	-.009	.001
$\Delta v$	<u>8.052</u>	<u>1.697</u>	<u>.284</u>	<u>.017</u>
120 days				
$\Delta x$ , km	-61 200	-13 100	-2220	13
$\Delta y$	11 600	2 800	783	271
$\Delta z$	-2 230	-441	-50	22
$\Delta R$	<u>62 300</u>	<u>13 400</u>	<u>2355</u>	<u>272</u>
$\Delta \dot{x}$ , m/sec	-8.452	-1.761	-.275	-.022
$\Delta \dot{y}$	-3.548	-.726	-.101	.020
$\Delta \dot{z}$	-.338	-.069	-.010	-.001
$\Delta v$	<u>9.173</u>	<u>1.906</u>	<u>.293</u>	<u>.030</u>
150 days				
$\Delta x$ , km	-83 000	-17 400	-2870	+90
$\Delta y$	- 1. 850	39	394	344
$\Delta z$	- 3 140	-629	-79	25
$\Delta R$	<u>83 100</u>	<u>17 401</u>	<u>2898</u>	<u>357</u>
$\Delta \dot{x}$ , m/sec	-7.346	-1.510	-.218	.035
$\Delta \dot{y}$	-6.700	-1.371	-.192	.037
$\Delta \dot{z}$	-.348	-.072	-.011	.001
$\Delta v$	<u>9.949</u>	<u>2.041</u>	<u>.291</u>	<u>.051</u>

TABLE 5.- MARS-VIKING RUN, DETAILED ACCURACY ANALYSIS  
FOR 180, 194, AND 199 DAYS

	5400	5401	5402	5403
180 days				
$\Delta x$ , km	-99 300	-20 700	-3 310	193
$\Delta y$	-20 700	-3 630	-65	497
$\Delta z$	-3 950	-792	-102	29
$\Delta R$	<u>101 500</u>	<u>21 030</u>	<u>3 312</u>	<u>534</u>
$\Delta \dot{x}$ , m/sec	-5.205	-1.055	-.135	.039
$\Delta \dot{y}$	-9.552	-1.981	-.271	.059
$\Delta \dot{z}$	-.299	-.063	-.009	.001
$\Delta v$	<u>10.882</u>	<u>2.245</u>	<u>.303</u>	<u>.071</u>
194 days				
$\Delta x$ , km	-104 300	-21 700	-3 450	319
$\Delta y$	-31 000	-5 800	-358	566
$\Delta z$	-4 040	-811	-105	30
$\Delta R$	<u>108 900</u>	<u>22 480</u>	<u>3 470</u>	<u>650</u>
$\Delta \dot{x}$ , m/sec	-5.102	-1.016	-.121	.046
$\Delta \dot{y}$	-14.473	-2.857	-.409	.062
$\Delta \dot{z}$	.779	.187	.026	-.007
$\Delta v$	<u>15.366</u>	<u>3.038</u>	<u>.427</u>	<u>.078</u>
199 days				
$\Delta x$ , km	247 500	289 000	-55 400	-10 900
$\Delta y$	-184 800	-170 100	-4 450	-3 530
$\Delta z$	-13 700	308 900	75 400	-6 030
$\Delta R$	<u>309 200</u>	<u>455 900</u>	<u>93 700</u>	<u>12 950</u>
$\Delta \dot{x}$ , m/sec	2962.6	2421.9	-422.8	-82.95
$\Delta \dot{y}$	-1182.4	-1324.8	-37.471	-35.13
$\Delta \dot{z}$	-24.51	2573.5	599.526	-475.01
$\Delta v$	<u>3189.9</u>	<u>3774.1</u>	<u>734.6</u>	<u>483.5</u>

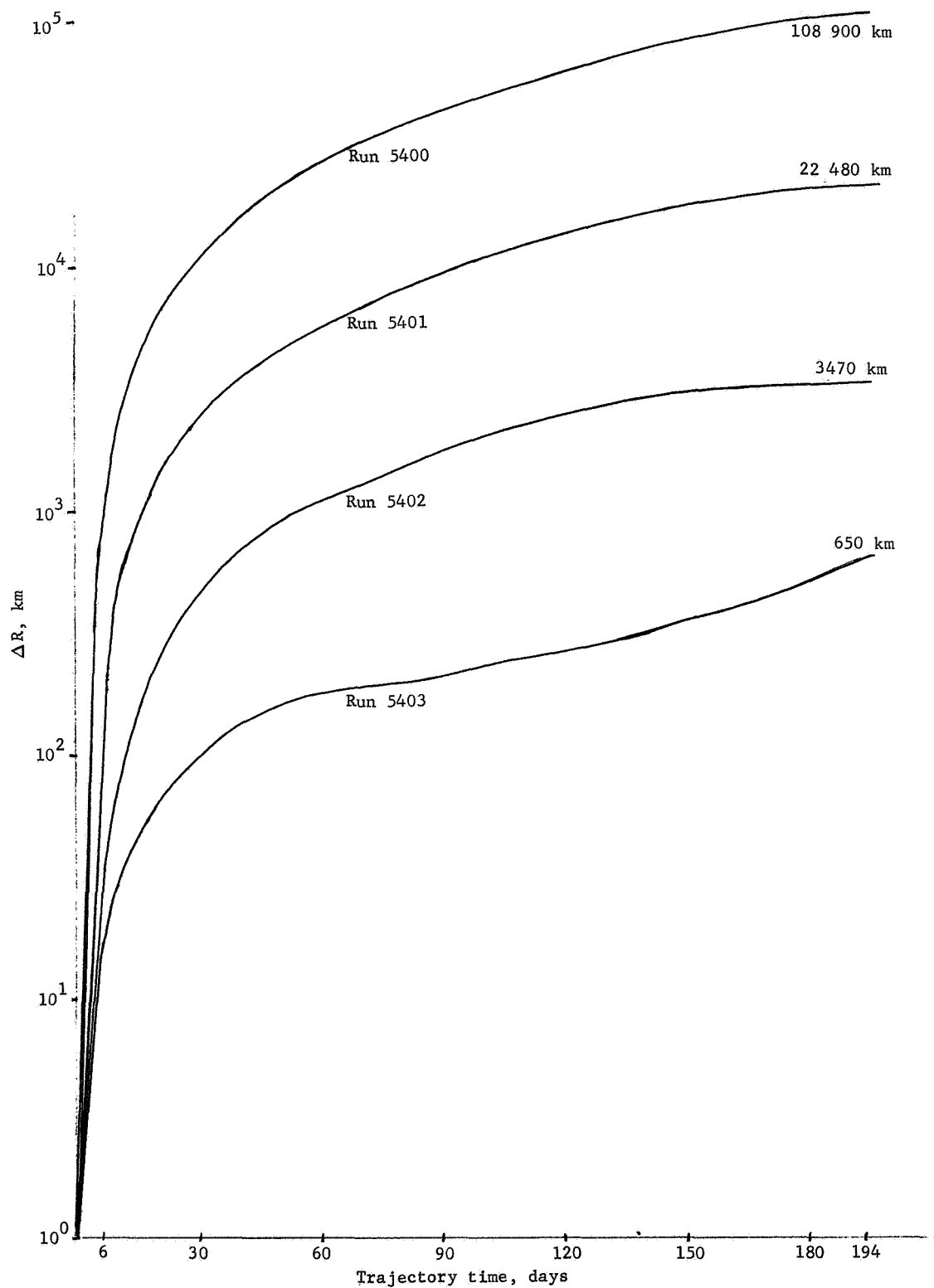


Figure 6.- Accuracy Analysis in Detail, Plot of  $\Delta R$  vs Trajectory Time

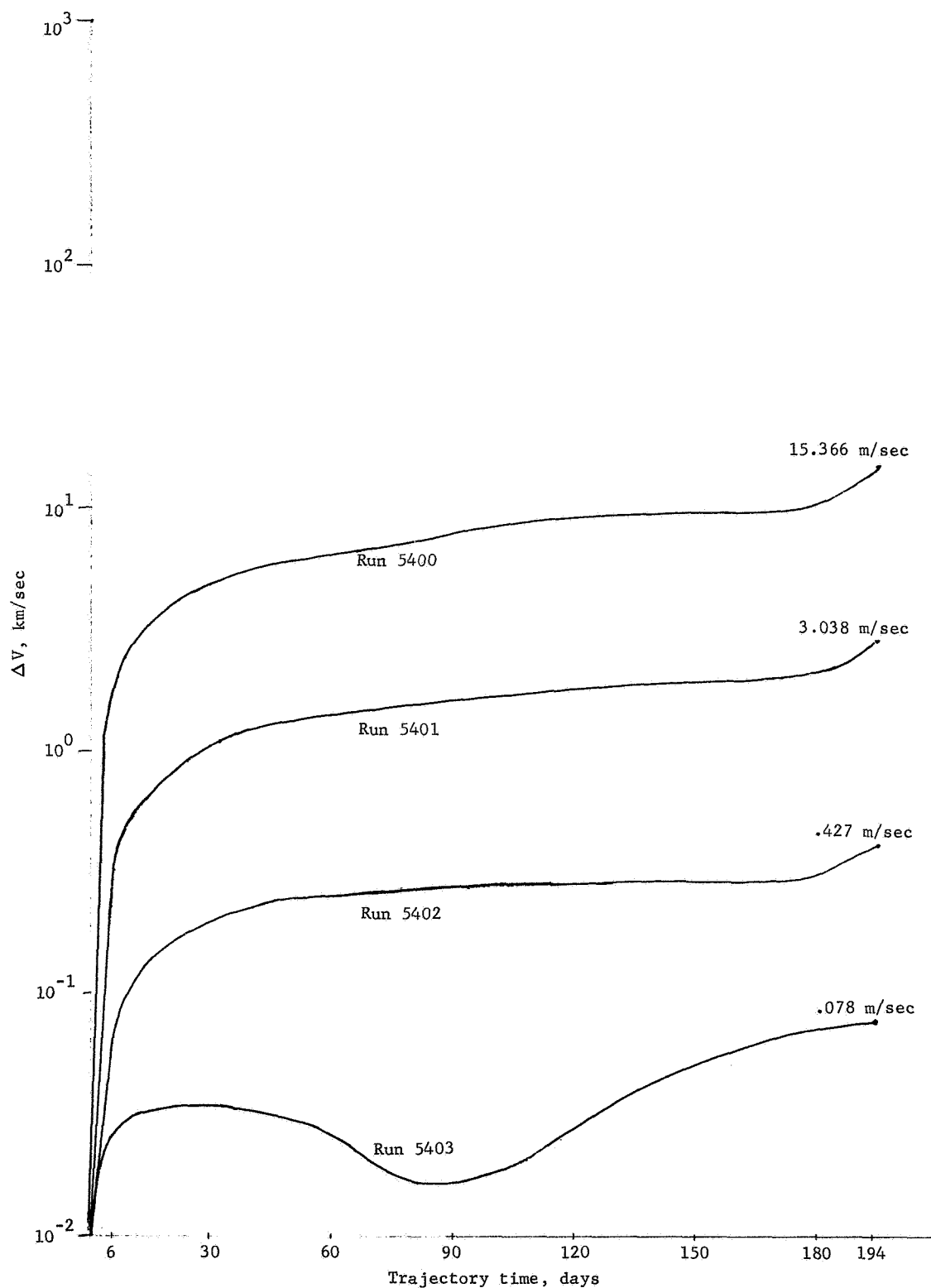


Figure 7.- Accuracy Analysis in Detail, Plot of  $\Delta V$  vs Trajectory Time

5403 at an accuracy level of  $1 \times 10^{-6}$ ) is different from the most accurate trajectory by only 15 km and 3 cm/sec after six days. This difference becomes 650 km and almost 8 cm/sec near the Martian sphere of influence.

Another striking feature of these trajectories is shown in table 5. It has long been an established fact in interplanetary trajectory analysis that swingby trajectories are incredibly delicate and highly sensitive. Readings from these accuracy studies were taken at 199 days, when the hypothetical spacecraft was about to leave the Martian sphere of influence and enter an orbit about the sun. The data shown comparing the various accuracy level runs with Run 5404 at 199 days confirms the almost amazing sensitivity of the swingby trajectory. Most important are the velocity discrepancies. Although Run 5403 (at accuracy level  $10^{-6}$ ) only differed from Run 5404 (at accuracy level  $2 \times 10^{-7}$ ) by 8 cm/sec going into Mars, its velocity difference coming out of Mars is an incredible 483 m/sec!

Figures 6 and 7, which were compiled using the data in tables 3, 4, and 5, demonstrate clearly the computational procession from the least accurate to the most accurate trajectory in the accuracy level study. These graphs do not include the rapid  $\Delta V$  increases near the planet Mars; the increases were too large for the scale.

The second kind of major study that was performed using the trajectory mode was an ephemeris analysis. The results of this study are summarized in table 6 and the locations of the various planets involved are shown in figures 8(a) and 8(b). For the best trajectory, also from the Mars-Viking launch window, the Earth's Moon was in a moderately favorable position; that is, the trajectory's closest approach to the Earth's Moon was at injection.

The purpose of the ephemeris analysis was to isolate the effect of each individual body in the solar system on a typical Viking trajectory. The results were startling. An additional purpose of the study was to ascertain the additional computer time required in using additional bodies for the ephemeris computation. The ground rules of the study should probably be explained before the results are interpreted. First, the same accuracy level, and hence the same step size, was used for all the runs. Second, exactly the same injection conditions, print out schemes, and interior logic were used for all the runs. In sum, the only difference between all the runs in the ephemeris analysis were the bodies included in the trajectory computations.



Table 6.- MARS-VIKING RUN, SUMMARY OF EPHEMERIS ANALYSIS

Run No.	Ephemeris description	Computer time sec,	B·T, km	B·R, km	$t_{SI}$ (a)	$R_{CA}$ , km	Based on Run 5402			
							$\Delta B \cdot T$ , km	$\Delta B \cdot R$ , km	$\Delta t_{SI}$ , days	$\Delta R_{CA}$ , km
5402	Earth, Mars, Sun, and Jupiter	93.011	7 521	550	90.1372	3 805	-----	-----	-----	-----
5002	+ Moon	64.499	-58 494	380 463	93.7776	379 891	-66 015	379 913	3.6408	376 086
5102	- Jupiter	67.581	-22 957	-369	90.1354	18 459	-30 478	-918	-.0018	14 654
5602	+ Saturn	<sup>b</sup> 85.128	5 188	-719	90.1260	b	-2 333	-1 269	-.0112	b
5502	+ Venus	109.005	8 113	1 942	90.1479	4 490	592	1 392	.0107	685
5702	+ Uranus	109.841	7 510	614	90.1378	3 802	-11	64	.0006	-3
5802	+ Neptune	110.158	7 536	588	90.1375	3 820	15	38	.0003	15
5302	+ Mercury	109.629	7 508	573	90.1374	3 796	-13	23	.0002	-9
5902	+ Saturn, Venus, Uranus, Neptune, and Mercury	<sup>b</sup> 137.632	5 763	824	90.1380	b	-1 758	274	.0008	b

<sup>a</sup>Last digits of Julian date

<sup>b</sup>Indicates impacted Mars

Note: Accuracy is  $5 \times 10^{-6}$

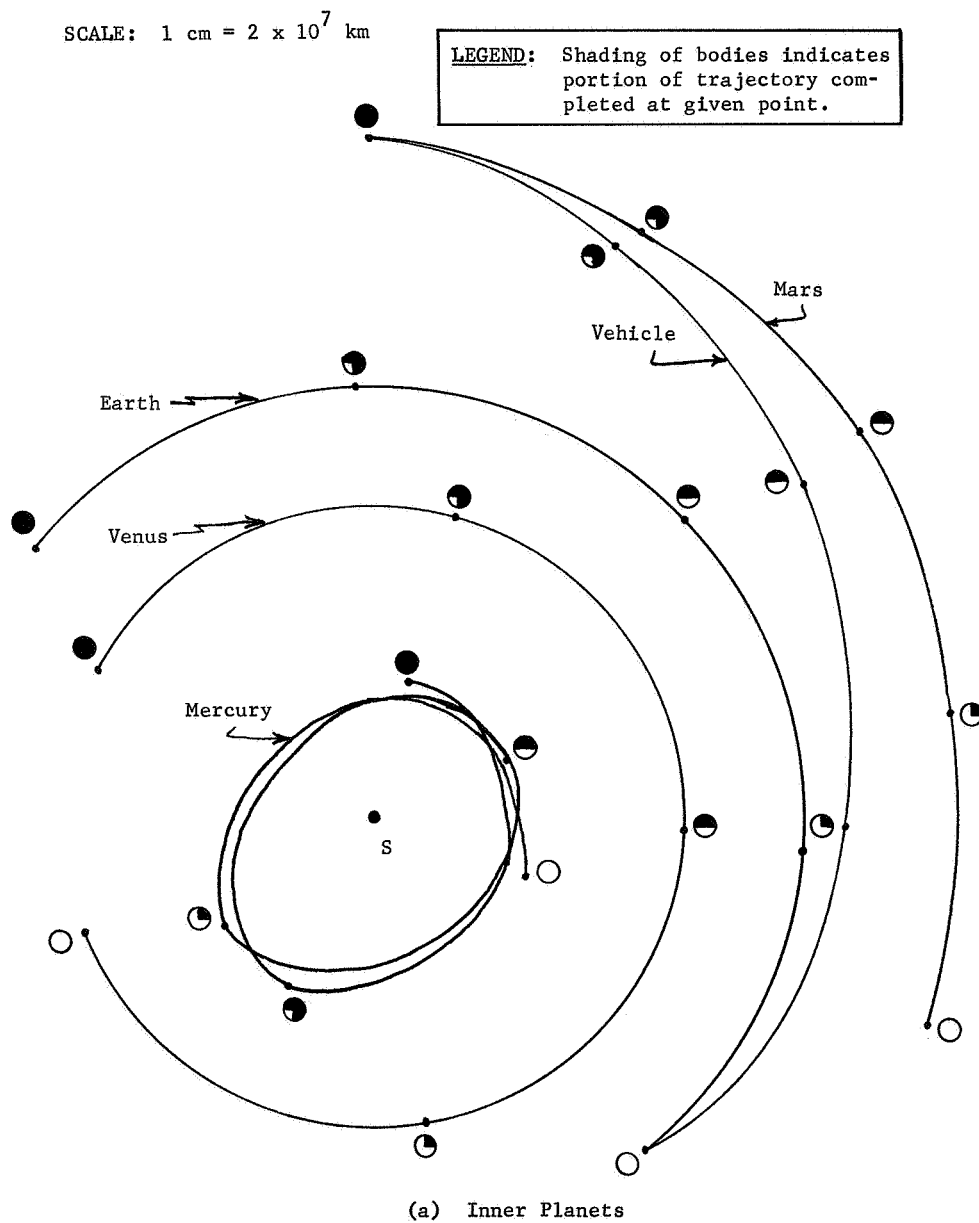





Figure 8.- Position of Planets during Test Trajectory


SCALE: 4 mm =  $10^8$  km

Saturn  


Mars  
  
S •

Jupiter  


Uranus  


Neptune  


(b) Outer Planets

Figure 8.- Concluded

A base run was made including, in the ephemeris, the Earth, the Sun, Mars, and Jupiter. The next run included the Earth's Moon together with the base bodies. A second run dropped out Jupiter, and considered only the Earth, the Sun, and Mars. The remainder of the runs included the Earth, the Sun, Mars, Jupiter, and one other planet (listed in table 6 under ephemeris description), except for Run 5902 which will be discussed in more detail later. The last four columns of the chart show the changes in key target-related quantities due to the inclusion or exclusion of each of the bodies. The results again verify the extreme sensitivity of interplanetary trajectories.

Outside of the Earth, Mars, and the Sun, table 6 demonstrates that the most important body on a Mars-Viking flight is the Earth's Moon. Its perturbation on the trajectory amounts to almost 400 000 km in B·R and over three and a half days in the trajectory time of flight to the Martian sphere of influence. It is important to note that for this study the Moon was on the opposite side of the Earth from the escape trajectory. One can immediately conjecture what a tremendous influence the Moon would have on a Martian flight passing even reasonably close to the Moon during escape.

In order of their magnitudes, the next most important bodies were shown to be Jupiter, Saturn, Venus, Uranus, Neptune, and Mercury. This order might well have been predicted, but the size of the perturbations was considered large. For example, both Saturn and Venus cause perturbations of over 1000 km in the B-plane quantities generally used for guidance. The three planets Uranus, Neptune, and Mercury, when their effects are combined, make over 100 km difference in B·R.

The question was raised, following the ephemeris analysis, whether or not the perturbations due to Uranus, Neptune, and Mercury could be regarded as computer noise. In an attempt to answer that question, Run 5902 was made with the base ephemeris plus the five extra planets that had previously been considered separately. The agreement between the resulting deviation from base Run 5402 and the sum of the deviations noted during each of the individual runs led to the conclusion that the perturbations on the trajectory accorded to each of the planets in the chart was accurate to within 10%. The cross-coupling between the effects of Saturn and Venus could also explain why the sums did not exactly agree with the results of Run 5902. The conclusions of the ephemeris analysis are, therefore, that the perturbations on a typical Mars-Viking run due to various other planets and bodies are essentially as given.

The computer time listed for each run should be clarified. For example, it took less computational time to calculate the trajectory with the Earth's Moon in the ephemeris than it did without it. On closer analysis, it is seen that the trajectory calculated with the Moon in the ephemeris did not pass very close to the surface of Mars, and hence not many small increments were needed to compute it. Similarly, Runs 5602 (with Saturn) and 5902 (with all the additional bodies) impacted Mars, thus stopping the integration. The most reasonable conclusion from the computer time column of table 6 is that a restricted five-body trajectory leaving Earth parking orbit and passing about 500 km above the surface of Mars, computed at an accuracy level of  $5 \times 10^{-6}$  and stopping at the outgoing Mars sphere of influence, requires about 93 sec of CDC 6500 computer time. The addition of an extra body adds approximately 16 sec to the time required for the computation.

Tables 7 and 8 give more detailed information about the ephemeris analysis. All the  $\Delta$  quantities shown are referenced to the base Run 5402 with Earth, Sun, Mars and Jupiter in the ephemeris. The charts show the perturbations due to each planet at selected trajectory epochs. The variations due to Uranus, Neptune, and Mercury, for example, are less than 1 km and about 1 mm/sec after six days of flight. These perturbations are magnified considerably by the state transition matrix over the flight. Figures 9 and 10 show the  $\Delta R$  and  $\Delta V$  perturbations due to each of the planets throughout the trajectory. The results appear to be more or less as expected except for the magnitude of the perturbations, as mentioned earlier.

A final comment from the ephemeris analysis is directed at the last set of numbers in table 8. Again all numbers are referenced to base Run 5402 except that now the trajectory time is 199 days, or roughly a day and a half after closest approach. The magnitudes of the deviations are quite large and serve as additional proof of the inordinate sensitivity of all ballistic swingby trajectories. Imagine, for example, that the trajectory was designed ballistically to continue on to Jupiter. The total integrated effects of even Uranus, Neptune, and Mercury on such a swingby trajectory would also be large.

The final trajectory mode study, which is summarized in table 9 and given in detail in table 10, was a targeted accuracy analysis study. The purpose of this study was to try to obtain the same target conditions for runs at a different accuracy level and

then observe the changes or deviations in the injection conditions. A proposed targeted ephemeris analysis study was discarded due to the expense of obtaining, from the targeting mode of the program, proper results to conduct the analysis. Recall that the earlier accuracy analysis study used the same injection conditions and then looked at target condition variations; the targeted study considers, for the different accuracy levels, the differences in injection conditions required to meet more or less the same target conditions.

The target conditions set up were  $B \cdot T = 9215$ ,  $B \cdot R = -990$ , and  $t_{SI} = 90.12497$ . Targeting runs at accuracies of  $5 \times 10^{-6}$  and  $2.5 \times 10^{-5}$  were considered acceptable when  $\Delta B \cdot T$  and  $\Delta B \cdot R$  were less than 75 km and  $t_{SI}$  was less than 0.0005 days. Greater tolerances were considered acceptable for the less accurate  $1.25 \times 10^{-4}$  run. The numbers of table 9 should therefore be compared with table 1 for complete understanding. It is clear from this comparison that each of the so-called "targeted" runs is close enough to permit interpretation of the injection condition difference as being due to the accuracy level difference.

The most interesting results from the targeted study are contained in the columns of table 9, which gives the  $\Delta$ -injection velocities compared to the most accurate run. For example, the injection velocities for a targeted run at  $5 \times 10^{-6}$  vary from those at  $1 \times 10^{-6}$  by less than 1/4 m/sec. Even the worst accuracy level,  $1.25 \times 10^{-4}$ , shows an injection velocity variation of about 5 m/sec for the worst components -- values that are less than 0.1% of the needed, geocentric ecliptic injection velocities. Table 10 shows detailed variations of the various targeted runs from the most accurate run.

The user of the trajectory mode must somehow analyze data similar to those presented and, for each use of the trajectory mode, determine its most efficient use. From the targeted accuracy analysis studies, it would seem that targeted run at a low accuracy level will produce a trajectory with the necessary characteristics for a detailed mission analysis. There is some question also whether or not using the trajectory mode in an extremely high accuracy  $2 \times 10^{-7}$  or even  $1 \times 10^{-6}$  makes much sense because of all the factors currently neglected in the n-body computation.

TABLE 7.- MARS-VIKING RUN, DETAILED EPHEMERIS ANALYSIS FOR 6, 60, AND 120 DAYS<sup>a</sup>

* (b)	+ Moon	- Jupiter	+ Saturn	+ Venus	+ Uranus	+ Neptune	+ Mercury
			6 days				
$\Delta x$ , km	-1 360	-25.6	0.5	-1.0	0.1	-0.2	0.4
$\Delta y$	-488	37.3	2.5	0.3	0.1	0.1	0.2
$\Delta z$	-301	0.1	0.2	0.1	0.1	-0.1	0.2
$\Delta R$	1 476	45.2	2.56	1.05	0.17	0.24	0.49
$\Delta \dot{x}$ , m/sec	-2.799	-1.075	.0021	-.0039	.0001	-.0005	.0005
$\Delta \dot{y}$	-.169	.1413	.0096	.0012	-.0001	-.0001	.0009
$\Delta \dot{z}$	-.747	.0011	.0003	.0001	.0003	-.0002	.0005
$\Delta v$	2.902	.1775	.0098	.0041	.0003	.00055	.0011
			60 days				
$\Delta x$ , km	-159 000	-3 098	-13	-128	-8	-5	-3
$\Delta y$	20 800	3 656	246	-3	-3	-3	9
$\Delta z$	6 300	45	-3	-1	1	-1	2
$\Delta R$	160 500	4 792	246	128	8.6	5.9	9.7
$\Delta \dot{x}$ , m/sec	-22.87	-1.266	-.011	-.059	-.007	-.005	-.006
$\Delta \dot{y}$	.59	1.254	.091	-.011	-.011	-.001	.002
$\Delta \dot{z}$	-.51	.012	-.002	-.002	.001	.001	.001
$\Delta v$	22.88	1.782	.092	.060	.007	.005	.0064
			120 days				
$\Delta x$ , km	-369 000	-11 800	64	358	-44	-22	-21
$\Delta y$	-51 300	10 500	988	-341	-17	-15	6
$\Delta z$	-16 800	85	-12	-34	1	-2	1
$\Delta R$	372 900	15 800	990	495.5	47	26.7	21.9
$\Delta \dot{x}$ , m/sec	-51.22	-1.696	.055	-.144	-.011	-.005	-.005
$\Delta \dot{y}$	-34.05	1.311	.218	-.151	-.005	-.004	-.003
$\Delta \dot{z}$	-2.60	.010	-.001	-.014	-.001	.001	-.001
$\Delta v$	61.56	2.144	.225	.209	.012	.006	.0059

<sup>a</sup>Referenced to Run 5402.

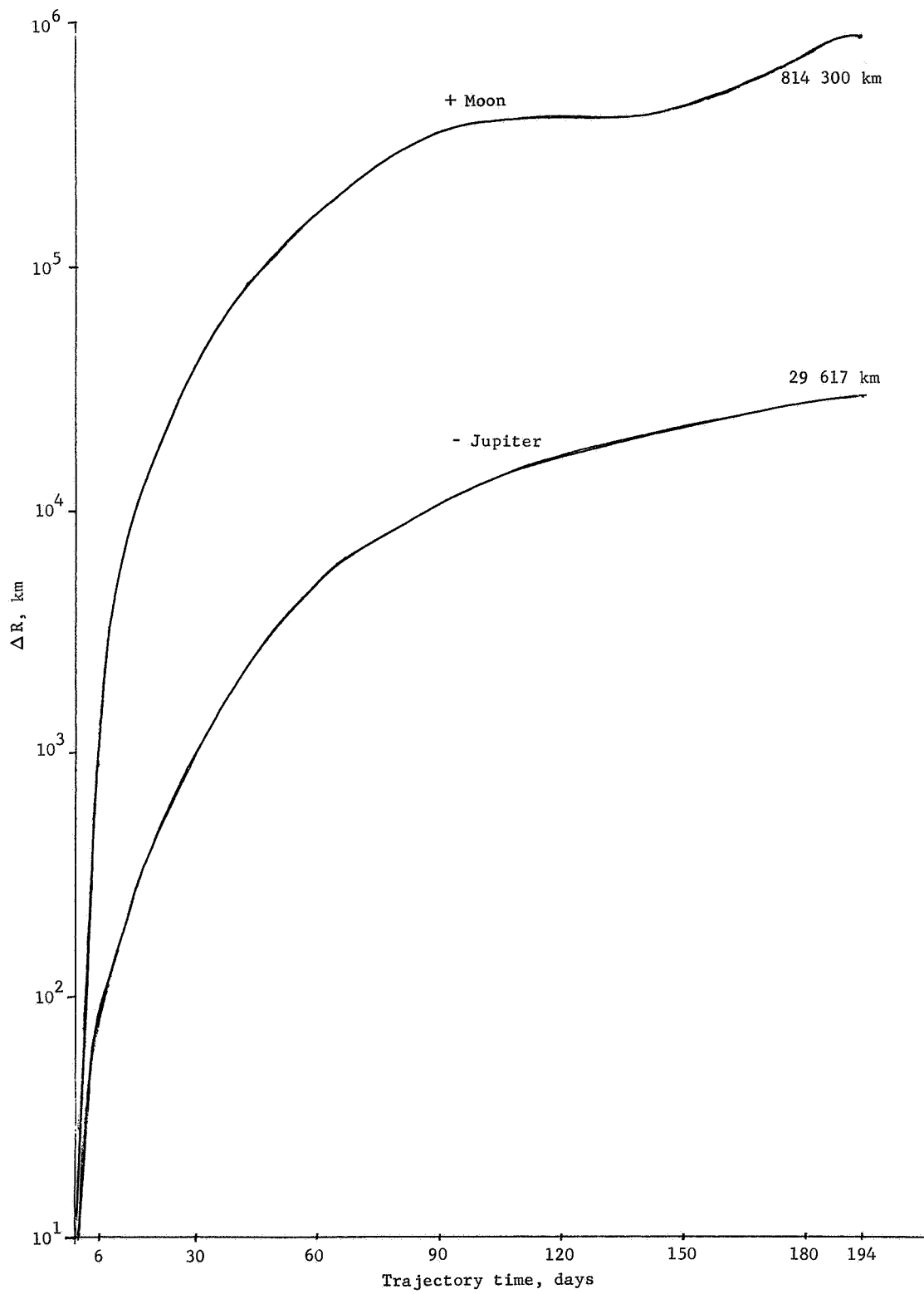
<sup>b</sup>Coordinates are all areocentric ecliptic.

TABLE 8.- MARS-VIKING RUN, DETAILED EPHEMERIS ANALYSIS FOR 180, 194, AND 199 DAYS<sup>a</sup>

(b)	+ Moon	- Jupiter	+ Saturn	+ Venus	+ Uranus	+ Neptune	+ Mercury
$\Delta x$ , km	-626 000	-18 300	180 days 609	-1 628	-108	-51	-47
$\Delta y$	-366 000	19 800	2959	-1 854	-72	-54	-17
$\Delta z$	-32 000	268	1	-142	-2	-2	1
$\Delta R$	725 000	26 900	3021	2 471	130	74	50
$\Delta \dot{x}$ , m/sec	-41.46	-7.36	.137	-2.210	-.013	-.006	-.004
$\Delta \dot{y}$	-86.45	2.617	.580	-.428	-.016	-.011	-.006
$\Delta \dot{z}$	-2.82	.068	-.005	-.023	-.001	.001	-.001
$\Delta v$	95.92	2.719	.596	.477	.021	.013	.007
$\Delta x$ , km	-669 000	-18 900	194 days 761	-1 849	-120	-56	-50
$\Delta y$	-463 000	22 800	3623	-2 338	-90	-67	-25
$\Delta z$	-33 900	347	2	-159	-2	-2	-1
$\Delta R$	814 300	29 600	3702	2 985	150	87	55.9
$\Delta \dot{x}$ , m/sec	-40.41	-3.74	.189	-.238	-.014	-.006	-.004
$\Delta \dot{y}$	-101.87	2.921	.700	-.514	-.020	-.014	-.009
$\Delta \dot{z}$	1.68	.076	-.031	.016	.001	.001	.001
$\Delta v$	109.61	2.946	.726	.567	.024	.015	.0099
$\Delta x$ , km	-316 000	199 days 1443 000	c	54 900	981	1 407	-24
$\Delta y$	-641 000	239 000	c	13 030	392	502	23
$\Delta z$	-149 000	115 000	c	32 900	2 847	1 212	1 260
$\Delta R$	730 000	516 000	c	65 316	3 037	1 924	1 260
$\Delta \dot{x}$ , m/sec	3 030.3	3 699.5	c	424.5	7.490	10.760	-.184
$\Delta \dot{y}$	-1 032.2	-2 035.1	c	135.8	4.167	4.841	.455
$\Delta \dot{z}$	-1 010.2	-918.9	c	269.5	22.748	9.717	10.035
$\Delta v$	3 356.9	4 321.1	c	520.8	24.309	15.285	10.047

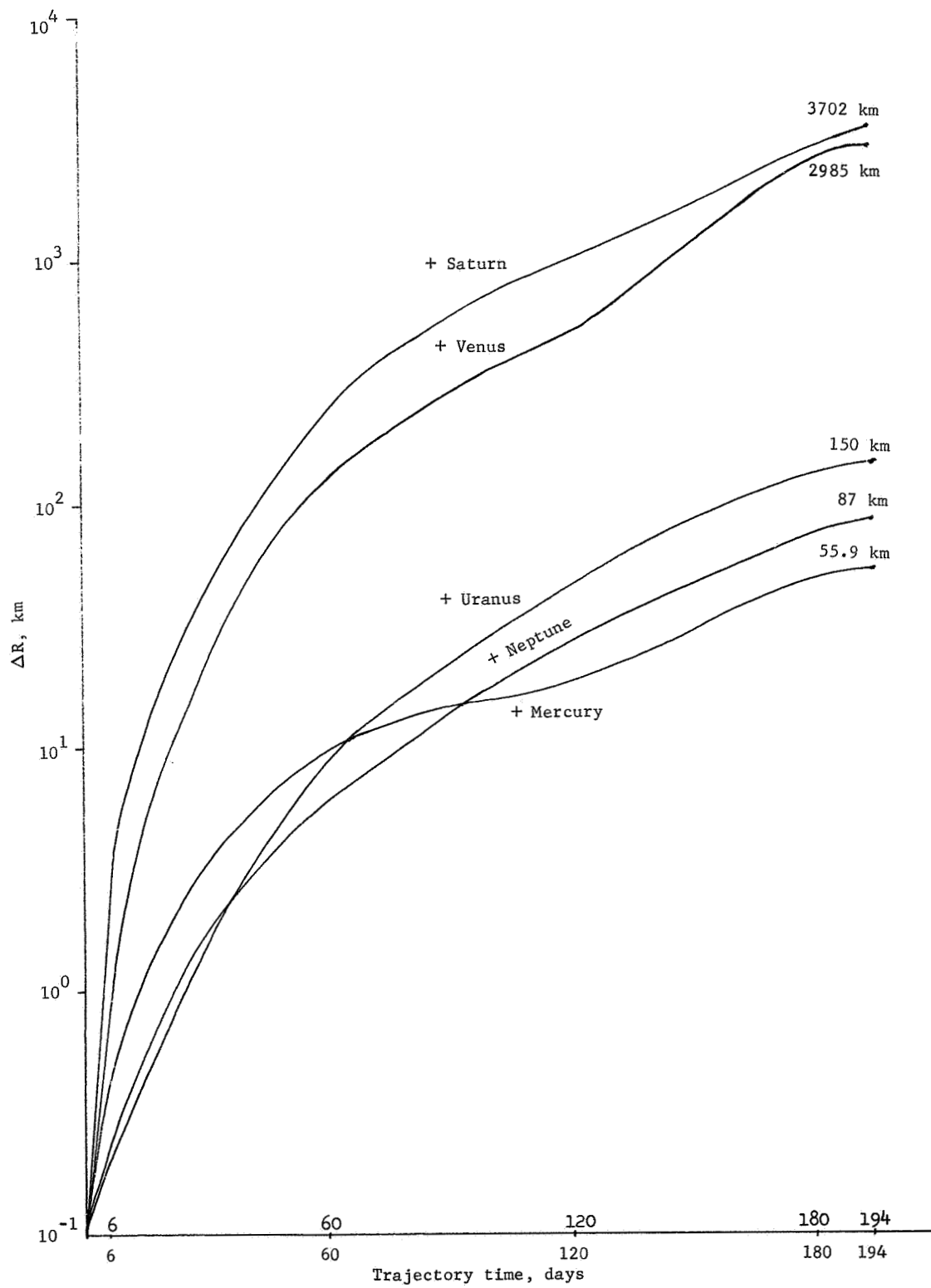
<sup>a</sup>Referenced to Run 5402.<sup>b</sup>Coordinates are all areocentric ecliptic.<sup>c</sup>Indicates impacted Mars.





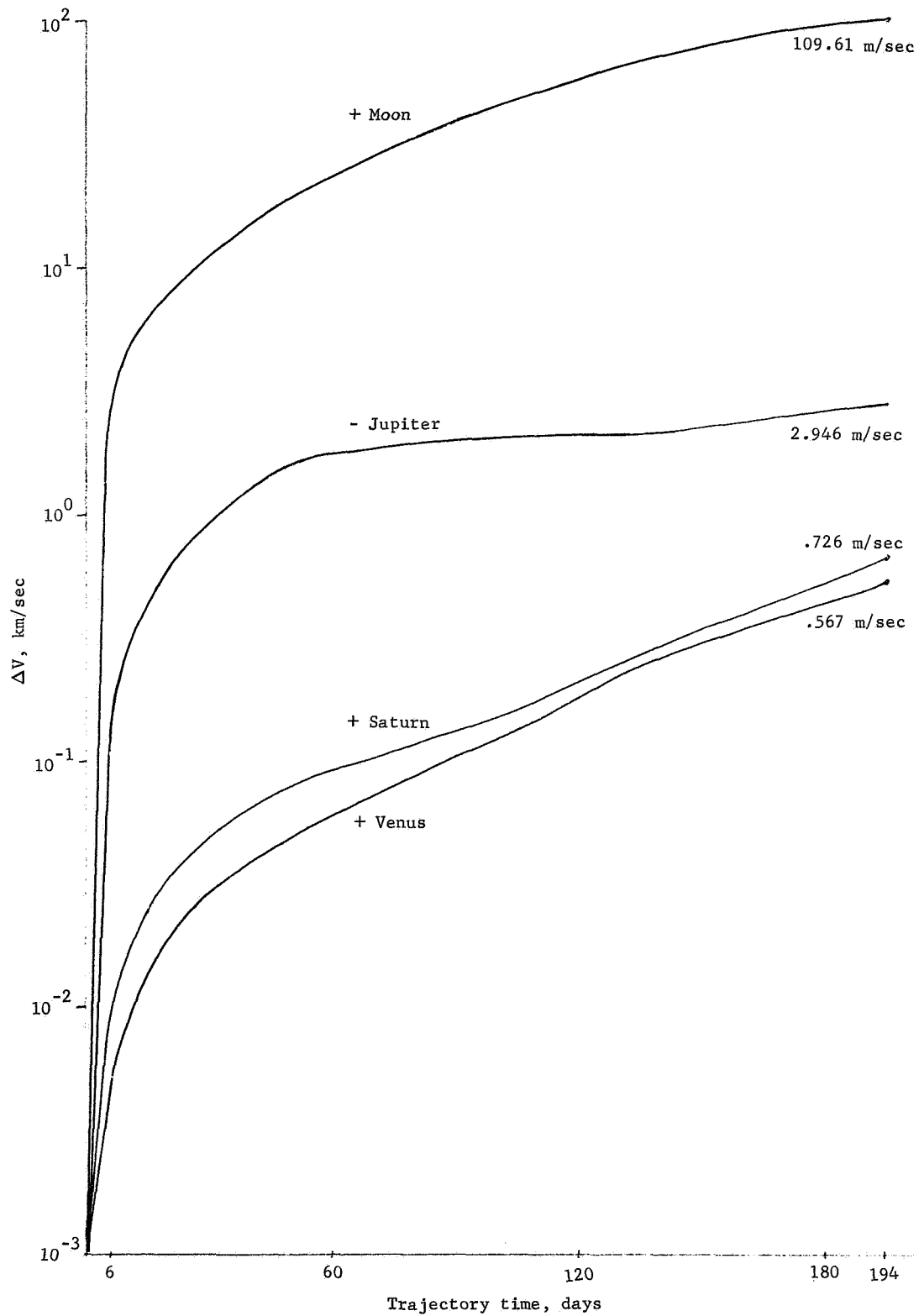
(a) + Moon, - Jupiter

Figure 9.- Ephemeris Analysis in Detail, Plot of  $\Delta R$  vs Trajectory Time



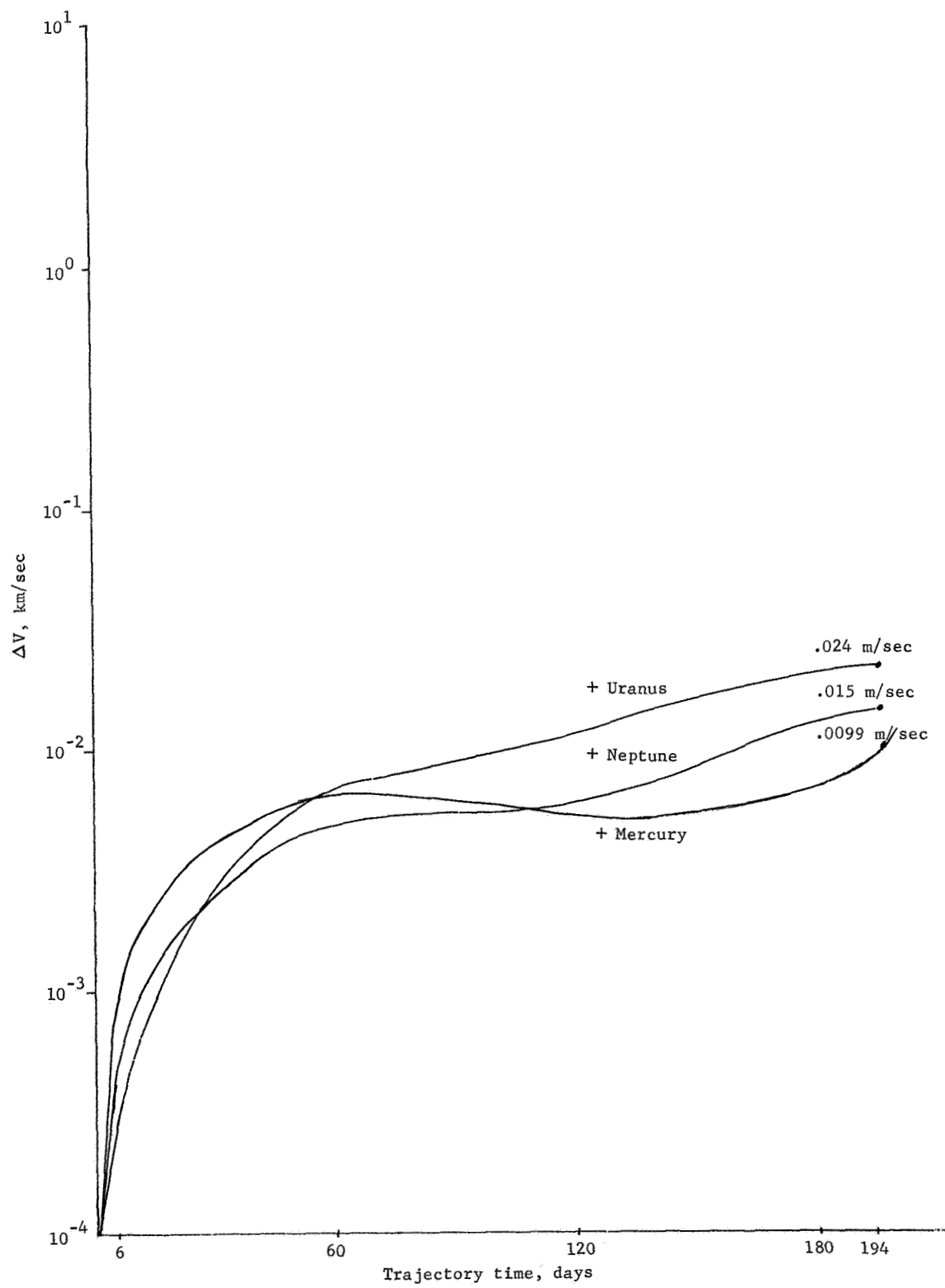
(b) + Saturn, + Venus, + Uranus, + Neptune, + Mercury

Figure 9.- Concluded



(a) + Moon, - Jupiter, + Saturn, + Venus

Figure 10.- Ephemeris Analysis in Detail, Plot of  $\Delta V$  vs Trajectory Time



(b) + Uranus, + Neptune, + Mercury

Figure 10.- Concluded

Table 9.- MARS-VIKING RUN, SUMMARY OF TARGETED ACCURACY ANALYSIS

Run No.	Accuracy level	Computer time, sec	Increment Number	Injection velocity, km/sec			Variations from 5453, m/sec			B.T, km	B.R, km	t <sub>SI</sub> (a)
				$\dot{X}_{GE}$	$\dot{Y}_{GE}$	$\dot{Z}_{GE}$	$\dot{\Delta X}$	$\dot{\Delta Y}$	$\dot{\Delta Z}$			
5450	$1.25 \times 10^{-4}$	21.434	964	9.522723	-3.472179	5.637579	1.517	-5.626	-5.252	9476	-1182	90.12355
5451	$2.5 \times 10^{-5}$	40.920	2046	9.521538	-3.467777	5.641676	.332	-1.224	-1.155	9521	- 999	90.12490
5452	$5 \times 10^{-6}$	82.492	4347	9.521259	-3.466780	5.642624	.053	-.227	-.207	9198	-933	90.12539
5453	$1 \times 10^{-6}$	170.895	9224	9.521206	-3.466553	5.642831	-----	-----	-----	9215	-990	90.12497
a <sub>L</sub> Last figures of Julian date												

TABLE 10.- MARS-VIKING RUN, DETAILED TARGETED  
ACCURACY ANALYSIS FOR 20, 100, AND  
197 DAYS

	5450	5451	5452
6 days			
$\Delta x$ , km	475	93	12
$\Delta y$	-689	-136	-24
$\Delta z$	152	21	1
$\Delta R$	851	166	27
$\Delta \dot{x}$ m/sec	.111	.023	.002
$\Delta \dot{y}$	-.173	-.033	-.006
$\Delta \dot{z}$	.086	.012	.001
$\Delta v$	.223	.042	.0064
100 days			
$\Delta x$ , km	199	18	37
$\Delta y$	-1170	-223	-38
$\Delta z$	470	62	2
$\Delta R$	1276	232	53
$\Delta \dot{x}$ , m/sec	.038	.005	-.007
$\Delta \dot{y}$	-.069	-.015	-.005
$\Delta \dot{z}$	.014	.002	.001
$\Delta v$	.080	.016	.0087
197 days			
$\Delta x$ , km	141.3	-0.4	-69.1
$\Delta y$	-711.7	-129.9	-164.8
$\Delta z$	308.6	49.6	48.2
$\Delta R$	788.5	139.0	185.1
$\Delta \dot{x}$ , m/sec	212.84	42.58	56.11
$\Delta \dot{y}$	-199.66	-25.98	-7.69
$\Delta \dot{z}$	-21.67	-2.91	4.57
$\Delta v$	292.63	49.96	56.82
<p><u>Note:</u> 1. All <math>\Delta</math> values are referenced to components of Run 5453 at same time.</p> <p>2. Coordinates are areocentric ecliptic.</p>			

## B. Targeting Mode

1. Introduction.- Numerous targeting problems were investigated during the design and construction of the targeting scheme. Typical examples will be given in this section to demonstrate both the scope and operation of the developed program.

A brief survey of the input data reveals the general range of applicability of the program. The required data includes the initial date, the target date, the gravitational bodies to be considered in the integration (including the launch and target planets), and the numerical differencing increment by which the velocity components are perturbed in computing the state transition matrix. The desired final accuracy level along with the intermediate accuracy levels to be used by the n-body subroutine are also specified by the user. The remaining options, which define the injection and target conditions, provide the capability for solving a wide range of problems.

Two options were allowed in the specification of the injection conditions. Generally, the injection position and zero-iterate injection velocity are computed internally from the mission constraints as described earlier. For this case the exact time of injection on the specified initial date is computed and the position and velocity are based on this corrected time.

A second injection condition option permits external specification of the exact injection time, position, and zero iterate velocity. In this option, the initial time is specified to thousandths of seconds, while in the first option it is generally prescribed only to a given day. The second option extends the analytical capability of the program. Midcourse corrections may be studied by specifying vehicle position and velocity at various times along a previously targeted trajectory and computing the velocity required to meet perturbed target conditions. In the same way one may compute the deflection velocity required by an entry vehicle ejected from a probe moving on a previously targeted swingby trajectory. This option also permits the efficient completion of partially targeted problems. Partially targeted problems may arise when the targeting algorithm generates a set of reasonably accurate injection conditions before encountering a region of strong nonlinearity where the previously computed state transition matrix is invalid. To target to within stringent tolerances on the target conditions in such a region, it may be necessary to reduce the velocity increment used in computing the state transition matrix.

There are four basic options permissible for the target conditions as well as two auxiliary options. The auxiliary options are quite similar. In the first auxiliary option, injection conditions are computed from a crude patched conic trajectory based on the launch date and planet and the target date and planet. These conditions, termed the point-to-point conditions, are generated with a bias which, while improving their effectiveness as initial values for n-body trajectories, degrades their validity in a patched conic propagation. For this reason a second auxiliary option is provided that calculates the unbiased patched conic injection conditions. These conditions would be a suitable zero iterate for determining targeted patched conics. Currently these patched conic conditions are never used internally by the program. Nevertheless the option is included in anticipation of the time that a patched conic propagation capability may be provided in the program.

Under the four basic targeting options, injection conditions result that are consistent with specific target conditions based on n-body trajectories. Option 3 includes the impact plane parameters  $B \cdot T$  and  $B \cdot R$  (targeted to selected tolerances) and an approximate time at sphere of influence  $t_{SI}$ . The fourth option targets to all three sphere of influence parameters. In option 5, the radius of closest approach  $r_{CA}$ , the inclination at closest approach  $i_{CA}$ , and time at closest approach  $t_{CA}$  are met approximately. In the sixth and final option of the targeting mode, all three closest approach conditions are satisfied to input tolerances. An analytical discussion of all these target conditions is provided in Chapter IV. The target conditions are summarized in table 11 for easy reference. Note that in option 5 the tolerances are given in terms of  $B \cdot T$ ,  $B \cdot R$ , and  $t_{SI}$  while the target conditions are  $i_{CA}$ ,  $r_{CA}$ , and  $t_{CA}$ . In this option the closest approach conditions are first converted to sphere of influence conditions  $B \cdot T$ ,  $B \cdot R$ , and  $t_{SI}$  and the iterative procedure is continued until these values are met to the tolerances  $\Delta B \cdot T$ ,  $\Delta B \cdot R$ , and  $\Delta t_{SI}$ . In option 6 the process proceeds in a similar manner but adds a step at the end to target to the specified closest approach tolerances  $\Delta i_{CA}$ ,  $\Delta r_{CA}$ , and  $\Delta t_{CA}$ .



TABLE 11.- SUMMARY OF TARGET OPTIONS

Option	Title	Required input
1	Point-to-point	$t_1, t_2, m_1, m_2$
2	Patched conic	$t_1, t_2, m_1, m_2$
3	2-variable SOI	$t_1, t_{SI}, m_1, m_2, \dots, m_n, ACC, B \cdot T, B \cdot R, \Delta B \cdot T, \Delta B \cdot R$
4	3-variable SOI	$t_1, t_{SI}, m_1, m_2, \dots, m_n, ACC, B \cdot T, \Delta B \cdot R, \Delta B \cdot T, \Delta B \cdot R, \Delta t_{SI}$
5	Approximate CA	$t_1, t_{CA}, m_1, m_2, \dots, m_n, ACC, i_{CA}, r_{CA}, \Delta B \cdot T, \Delta B \cdot R, \Delta t_{SI}$
6	Strict CA	$t_1, t_{CA}, m_1, m_2, \dots, m_n, ACS, i_{CA}, r_{CA}, \Delta i_{CA}, \Delta r_{CA}, \Delta t_{CA}$

2. Targeting to closest approach conditions (target options 5 and 6)..- Since a target option 6 problem essentially demonstrates the features of all the targeting options and completely includes a target option 5 problem this example will be discussed in detail. Tables 12 and 13, which supply the actual computer output for the example problem, will be used as references.

Table 12 summarizes the preliminary work of the targeting algorithm. The general mission constraints are:

Launch date: 7/24/73;

Target date: 2/16/74;

Gravitational bodies (5): Sun, Earth (launch planet), Mars (target planet), Jupiter, Moon.

The injection option is set so that the injection position and zero-iterate velocity are computed internally in the program using the point-to-point conditions. The target conditions and tolerances are specified (option 6) as:

$$\begin{array}{ll}
 i_{CA} = 38^\circ & \Delta i_{CA} = 0.25^\circ \\
 r_{CA} = 4800 \text{ km} & \Delta r_{CA} = 25 \text{ km} \\
 t_{CA} = 2/16/74 & \Delta t_{CA} = 0.001 \text{ day}
 \end{array}$$



TABLE 13.- ACTUAL TARGETING PROCEDURE

## NUMERICAL DIFFERENCING PROCEDURE

L I S A X Y Z TRAJECTORY TRAJECTORY TRAJECTORY TARGET TARGET TARGET TIME TOTAL NO	V T T C D O D S.T B.R TSI B.T/INCL B.R/RCA TSI/TCA PER CP OF	E E E C O O O OR OR OR STATE TRANSITION MATRIX INTEG TIME INTRG	L R P Y T T T
---	--	---	---------------

## TARGETING AND CONSTRUCTION OF SPHERE-OF-INFLUENCE STATE TRANSITION MATRIX

1 0 0	5.00E-04	34.386674	12.518986	6.164171	-160137.77	122307.69	27072.827	5074.68	6930.76	27072.348	5.86	6.1	253
1 0 1	5.00E-04	34.386684	12.518986	6.164171	-159992.04	121656.42	27072.822	-1.25E-07	5.31E-07	-7.92E-02			
1 0 2	5.00E-04	34.386674	12.518996	6.164171	-160251.97	122573.48	27072.829	-1.87E-07	8.06E-09	-6.88E-03			
1 0 3	5.00E-04	34.386674	12.518986	6.164181	-160046.14	121883.85	27072.824	7.48E-08	-8.35E-07	1.17E-01			
1 1 0	5.00E-04	34.342614	12.490438	6.216694	-20272.15	148999.45	27073.464	5112.11	6945.06	27072.274	5.97	29.7	258
1 1 1	5.00E-04	34.342654	12.490438	6.216694	-20132.54	148326.10	27073.459	-3.23E-08	5.29E-07	-7.74E-02			
1 1 2	5.00E-04	34.342614	12.490448	6.216694	-20382.10	149259.00	27073.466	-1.76E-07	3.06E-09	-5.55E-03			
1 1 3	5.00E-04	34.342614	12.490438	6.216704	-20186.61	148574.29	27073.461	-5.60E-08	-8.59E-07	1.19E-01			
1 2 0	5.00E-04	34.358787	12.492152	6.195468	10751.43	744.36	27072.307	5099.87	6929.76	27072.256	6.28	53.9	271
1 2 1	5.00E-04	34.358797	12.492152	6.195468	10917.11	92.12	27072.303	-1.58E-08	4.20E-07	-7.34E-02			
1 2 2	5.00E-04	34.358787	12.492162	6.195468	10631.94	994.84	27072.308	-1.74E-07	-3.27E-09	-6.84E-03			
1 2 3	5.00E-04	34.358787	12.492152	6.195478	10851.45	337.37	27072.304	-8.19E-08	-7.00E-07	1.13E-01			
1 3 0	5.00E-04	34.365213	12.493463	6.185825	4867.09	6621.97	27072.254	5098.80	6930.70	27072.260	6.28	79.1	271
1 3 1	5.00E-04	34.365223	12.493463	6.185825	5031.29	5972.44	27072.250	-1.93E-08	4.25E-07	-7.34E-02			
1 3 2	5.00E-04	34.365213	12.493473	6.185825	4744.13	6880.87	27072.256	-1.70E-07	-5.92E-09	-6.10E-03			
1 3 3	5.00E-04	34.365213	12.493463	6.185835	4966.08	6218.22	27072.252	-7.78E-08	-7.12E-07	1.14E-01			
1 4 0	5.00E-04	34.364914	12.493387	6.186248	5076.49	6968.35	27072.260	5098.90	6930.74	27072.259	6.27	104.3	271

## TARGETING TO SPHERE-OF-INFLUENCE CONDITIONS

2 0 0	2.50E-05	34.364914	12.493387	6.186248	122071.78	-187536.25	27071.252	4897.54	6867.20	27072.439	23.90	128.2	1035
3 0 0	5.00E-06	34.362602	12.504879	6.192522	28737.57	-8840.80	27072.152	4947.03	6895.41	27072.282	50.51	178.7	2193
3 1 0	5.00E-06	34.360244	12.508034	6.197948	8416.10	3707.69	27072.259	4950.82	6896.29	27072.300	50.54	229.3	2193
3 2 0	5.00E-06	34.358656	12.508351	6.200676	5686.19	6244.80	27072.296	4952.16	6896.57	27072.301	50.53	279.8	2193
3 3 0	5.00E-06	34.358536	12.508440	6.200861	5067.57	6829.07	27072.300	4952.34	6896.64	27072.301	50.57	330.4	2193
3 4 0	5.00E-06	34.358546	12.508457	6.200855	4962.45	6893.56	27072.301	4952.35	6896.64	27072.301	50.57	381.0	2193

## CONSTRUCTION OF CLOSEST-APPROACH STATE TRANSITION MATRIX

1 0 0	5.00E-04	34.364914	12.493387	6.186248	37.05	4887.46	27074.501	38.00	4800.00	27074.500	9.28	390.3	402
1 0 1	5.00E-04	34.364915	12.493387	6.186248	36.74	4862.01	27074.500	6.16E-06	9.31E-08	-1.17E-02			
1 0 2	5.00E-04	34.364914	12.493388	6.186248	37.21	4898.07	27074.501	2.34E-05	-1.01E-08	-1.56E-02			
1 0 3	5.00E-04	34.364914	12.493387	6.186249	36.81	4860.25	27074.501	3.35E-06	-1.28E-07	4.85E-03			

## TARGETING TO CLOSEST-APPROACH CONDITIONS

1 0 0	5.00E-06	34.358546	12.508457	6.200855	37.02	4817.95	27074.500	38.00	4800.00	27074.500	78.33	496.5	3399
1 1 0	5.00E-06	34.358546	12.508475	6.200862	38.11	4846.91	27074.500	38.00	4800.00	27074.500	78.31	574.9	3398
1 2 0	5.00E-06	34.358547	12.508480	6.200865	37.94	4788.42	27074.500	38.00	4800.00	27074.500	78.30	653.2	3400

A typical targeting schedule is used. Thus the initial targeting and construction of the state transition matrices is done at an integration accuracy level of  $5 \times 10^{-4}$ , one iteration is included at  $2.5 \times 10^{-5}$  to efficiently reduce the error between the first and third accuracy levels and the final accuracy level of  $5 \times 10^{-6}$  is the level at which the trajectory is to be ultimately targeted. No more than five iterations are to be made at this final (costly) accuracy level. The basic velocity increment used in the construction of the state transition matrix is  $1 \times 10^{-5}$  km/sec. The targeted trajectory position and velocity will be printed at intervals of every 100 integration increments and 500 flight days.

The point-to-point conditions (option 1) are generated to compute the injection conditions. The injection time of 9 hr 25 min 47.639 sec is based on a launch from Cape Kennedy on 7/24/73 at 8 hr 16 min 16 sec (when Cape Kennedy is at a required position) to an intermediate parking orbit of 100 n. mi. The injection position and zero-iterate velocity corresponding to this injection time is then computed. The target conditions  $i_{CA}$ ,  $r_{CA}$ ,  $t_{CA}$  are then converted to equivalent conditions  $B \cdot T$ ,  $B \cdot R$ , and  $t_{SI}$  at the Mars sphere of influence using the point-to-point conditions to approximate the approach asymptote.

The actual targeting procedure, depicted in table 13, may now be initiated. The integration of the point-to-point conditions at the first accuracy level yields a trajectory intersecting the Martian sphere of influence; hence no "outer targeting" is required. The approach asymptote of that trajectory is used in conjunction with the desired closest approach conditions  $i_{CA}$ ,  $r_{CA}$ , and  $t_{CA}$  to recompute the auxiliary target conditions  $B \cdot T$ ,  $B \cdot R$ , and  $t_{SI}$ . The actual trajectory parameter values differed from these target values by  $\Delta B \cdot T = -165\,000$  km,  $\Delta B \cdot R = 115\,000$  km, and  $\Delta t_{SI} = 0.5$  day. The state transition matrix computed about this nominal is then used to predict an improved injection velocity. This velocity is subsequently the basis for the generation of a new trajectory, the auxiliary target conditions are recomputed using the new approach asymptote, and the actual trajectory parameter values are calculated. The process is repeated until, after four iterations, a targeted trajectory is determined at the low accuracy level with target errors of  $\Delta B \cdot T = 22$  km,  $\Delta B \cdot R = 38$  km,

$\Delta t_{SI} = 0.001$  day. The injection velocity used in generating this trajectory is then used at the second integration accuracy level, the auxiliary target values computed, and the actual trajectory values recorded. The final state transition matrix generated at the first accuracy level is then used to predict a corrected injection velocity vector. This velocity is then sent to the third accuracy level and the entire targeting process is repeated, always using the same state transition matrix constructed on the last step at the first accuracy level. The fourth iterate of that process has errors of  $\Delta B \cdot T = 10$  km,  $\Delta B \cdot R = 3$  km, and  $\Delta t_{SI} < 0.001$  day. If option 5 had been chosen, the targeting would be finished. The resulting trajectory integrated forward to closest approach would have target errors of  $\Delta i_{CA} = 0.98^\circ$ ,  $\Delta r_{CA} = 18$  km, and  $\Delta t_{CA} < 0.001$  day.

Since this is an option 6 targeting problem, the program now takes the final injection velocity generated at the first accuracy level and integrates this to closest approach, computing the trajectory parameter values at that point. A state transition matrix is constructed about this trajectory, now relating changes in the closest approach target conditions to changes in the injection velocity. The return is then made to the third accuracy level where the final injection velocity generated at that level is integrated to closest approach. The closest-approach state transition matrix just constructed is now used repeatedly to target the trajectory. In two iterations target errors of only  $\Delta t_{CA} = 0.06^\circ$ ,  $\Delta r_{CA} = 12$  km, and  $\Delta t_{CA} < 0.001$  are attained.

It is interesting to compare the computer times required by each of the different types of integrations used in this problem. The data are summarized in table 14. The type 1 integrations are required most often (17 integrations), but they are the cheapest in computer time. The number of the most time consuming integrations (type 5) is held to a minimum. The advantage of using the lower accuracy integrations is obvious. A brute force numerical differencing technique using only type 5 integrations would require 1576 sec of integration computer time if but five iterations were needed for the targeting; the modified approach using many more iterations requires 655 sec.

TABLE 14.- DISTRIBUTION OF CDC 6500 COMPUTER TIME  
IN TYPES OF INTEGRATIONS

Type	Final boundary	Accuracy	Integration steps	Number of integrations required	Computer time per integration, sec	Total computer time, sec
1	Sphere of influence	$5 \times 10^{-4}$	271	17	6.28	106.76
2	Sphere of influence	$2.5 \times 10^{-5}$	1035	1	23.90	23.90
3	Sphere of influence	$5 \times 10^{-6}$	2193	5	50.55	252.75
4	Closest approach	$5 \times 10^{-4}$	402	4	9.28	37.12
5	Closest approach	$5 \times 10^{-6}$	3400	3	78.30	234.90

Finally, the results of the fifth and sixth targeting options as applied to this problem may be compared. The option 5 injection conditions were generated in 383 sec of CDC 6500 computer time (requiring only the first three integration types). An additional 272 sec is needed to target to the option 6 conditions. This great expense in computer time required by the strict closest approach targeting prompted the inclusion of the relaxed conditions available under option 5. A summary of the two options is given in table 15.

TABLE 15.- COMPARISON OF TARGET OPTIONS 5 AND 6

Option	Injection velocity, km/sec			Trajectory target values		
	$V_x$	$V_y$	$V_z$	$i_{CA}$ , deg	$r_{CA}$ , km	$t_{CA}$ , error in days
5	34.358546	12.508457	6.200855	37.02	4817.95	0.001
6	34.358547	12.508480	6.200865	37.94	4788.42	0.001

As noted in Chapter IV, there are four distinct trajectories that have identical inclinations, radii, and times at closest approach for a given mission. The example discussed immediately above is therefore only one of four possible solutions to the problem. A second posigrade orbit having an inclination of  $38^\circ$  is specified by setting the target inclination equal to

-38°; two retrograde orbits are determined by the inclination values  $\pm 142^\circ$  (plus or minus the supplement of the desired inclination). The characteristics of this family of solutions are summarized in table 16. The results given are from trajectories targeted at option 5 at an accuracy level of  $1.25 \times 10^{-4}$  using the Sun, Earth, and Mars as gravitational bodies. The trajectories again demonstrate the validity of the target option 5 approximations.

TABLE 16.- FAMILY OF SOLUTIONS TO CLOSEST APPROACH PROBLEM

Input $i_{CA}$ , deg	Injection velocity, km/sec			Auxiliary conditions			Target errors		
	$V_x$	$V_y$	$V_z$	B·T, km	B·R, km	$t_{ST}$	$i_{CA}$ , deg	$r_{CA}$ , km	$t_{CA}$
+38	34.366621	12.508799	6.187236	4957	6900	2/13/74	-0.89	35	+0.001
-38	34.362456	12.508296	6.193847	7994	-2872	2/13/74	-0.22	99	+0.000
+142	34.365151	12.511210	6.191176	-7994	2873	2/13/74	-0.38	-163	+0.000
-142	34.360912	12.510700	6.197898	-4955	-6897	2/13/74	-0.77	-122	+0.000

3. Targeting to sphere of influence conditions.- A Venus probe will be used to illustrate the differences between target options 3 and 4. The general problem parameters are given below:

Injection date: 11/3/73;

Target date: 2/20/74 (at sphere of influence);

Gravitational bodies: Sun, Earth (launch planet), Venus (target planet), Jupiter, Moon;

Final accuracy level:  $2.5 \times 10^{-5}$ ;

Intermediate level:  $5.0 \times 10^{-4}$ ;

Target conditions: B·T = 3349,

B·R = 1689,

$t_{SI}$  = 2/20/74.

For targeting in option 3, the above time constraint on  $t_{SI}$  is not rigidly enforced. This time is used, however, to generate the injection conditions, which results in the final time being satisfied approximately. The targeting for both problems begins with the internally generated point-to-point injection conditions.

The progressive targeting under the two options is summarized in table 17. In option 3, 166 sec are required to target to B·T and B·R. In adding the time constraint in option 4, the required computer time is increased to 251 sec. This time differential is produced by two effects. First, in option 3 the construction of the 2 x 2 state transition matrix computed at each iteration in the first accuracy level requires two integrations, while in option 4 that matrix is 3 x 3 and so necessitates three integrations. Secondly, more iterations are used in targeting option 4.

TABLE 17.- COMPARISON OF TARGET OPTIONS 3 AND 4

		Target option 3				Target option 4			
Level 1		B·T, km	B·R, km	$t_{SI}(a)$	Computer time, sec	B·T, km	B·R, km	$t_{SI}(a)$	Computer time, sec
		399 966	-99 684	77.745	6.1	399 966	-99 684	77.745	6.1
		-8 017	10 468	78.114	23.7	213 802	-119 630	78.342	29.9
		3 971	1 315	78.068	41.2	55 072	-29 862	78.453	53.8
						25 805	-10 289	78.445	77.6
						3 942	1 378	78.499	101.4
Level 2		-165 080	36 985	78.394	75.3	-164 867	38 542	78.832	136.1
		-18 070	13 508	78.164	98.0	43 304	-23 782	78.300	158.8
		3 050	1 845	78.100	120.7	-12 807	11 030	78.552	181.5
		3 323	1 704	78.099	143.3	4 656	878	78.493	204.2
		3 367	1 678	78.099	165.9	2 794	2 019	78.502	227.0
						3 445	1 638	78.500	250.8
		$V_x$ , km/sec	$V_y$ , km/sec	$V_z$ , km/sec		$V_x$ , km/sec	$V_y$ , km/sec	$V_z$ , km/sec	
		-22.36441	11.50466	<sup>b</sup> 2.76032		-22.43800	11.54728	2.85894	
<sup>a</sup> Last figures of Julian date referenced to 1900. Target date = 27078.500.									
<sup>b</sup> Z-component of injection velocity is constrained to this value under option 3 strategy.									



If the final time constraint does not need to be met exactly, targeting option 3 obviously offers a more efficient solution than does option 4. However, if the time constraint must be satisfied, option 4 can be used.

4. Computational procedures for difficult problems.- There are two procedures added to the basic numerical differencing algorithm to handle certain irregularities. It is possible that the point-to-point injection conditions generated by the program will lead to a trajectory which does not intersect the target planet sphere of influence. The outer targeting scheme discussed in Chapter IV is activated automatically when this contingency occurs. A typical example of outer targeting is depicted in table 18. The basic mission under investigation is an Earth-Mars probe leaving on 7/31/73 and arriving at the sphere of influence of Mars on 2/11/74. Additional gravitational bodies included in the n-body trajectory generation are the Sun, Moon, and Jupiter. The first accuracy level for this problem was  $1.25 \times 10^{-4}$ . Upon integration of a set of erroneous injection conditions, the trajectory had a closest approach to Mars of 2.5 million km and arrived ten days earlier than desired. An "artificial" sphere of influence was constructed about Mars having a radius of 3 million km. "Artificial" target conditions of  $B \cdot T = B \cdot R = 0$  and a biased time were used as constraints in an option 4 targeting process. In five iterations injection conditions were generated which led to a trajectory intersecting the Martian sphere of influence with a time error of one day. These conditions were then subsequently targeted to the desired constraints.

The second protective measure is called a bad-step check. In this procedure each iterate is compared with its predecessor before being accepted. If the new iterate is considered inferior to the old, the velocity correction is reduced by one-fourth and a new iterate computed. The process is then continued until an improved iterate is determined. There are two criteria for deeming an iterate inferior. The first occurs when a new iterate misses the target planet sphere of influence while the preceding iterate intersected it. The second criterion involves a scalar measure of the error that is assigned to each iterate. The error measure is arbitrarily defined as  $E = |B \cdot T| + |B \cdot R| + 100000$   $|t_{SI}|$  or  $E = 100 |i_{CA}| + |r_{CA}| + 10000 |t_{CA}|$  depending on the target conditions. When the error measure is increased by any iterate, that iterate is termed inferior.

TABLE 18.- DEMONSTRATION OF OUTER TARGETING

Iteration	B·T, km	B·R, km	$t_{SI}$ , last figures of Julian date
Outer targeting loop:			
0	1 033 217	-1 906 363	72.707
1	997 753	-41 205	88.591
2	423 538	-1 131 272	77.353
3	100 219	257 898	85.825
4	-20 417	125 533	83.896
5	1 079	2 955	83.047
Outer targeting conditions	0	0	83.015
Return to regular targeting:			
0	10 402	6 630	91.378
Regular target conditions	8 000	1 000	90.125

The need for a bad-step check is demonstrated in the example illustrated in table 19. The mission constraints included departure from Earth on 3/9/72 and arrival at the sphere of influence of Venus on 9/5/72. The gravitational effects of the Sun, Earth, Venus, Jupiter, and the Moon were to be considered in the integration. Table 19 indicates the outer targeting (at the  $5.0 \times 10^{-4}$  accuracy level) for the problem. The trajectory is obviously in an extremely nonlinear region in terms of the targeting procedure, and in fact is the worst case encountered to date. Each of the first two corrections lead to trajectories missing the artificial sphere of influence (having a radius of 1 065 496 km). Subsequently three sets of velocity corrections lead to target conditions that increase the error measure assigned to the previous iterate. This example emphasizes the necessity for including the bad-step check; if it were not present the targeting would fail. It furthermore points up the advantage of performing the early targeting at a low accuracy level. In spite of the pathological nature of this problem, the outer targeting required only 220 sec of computer time to generate injection conditions.

that intersected the target planet sphere of influence with  $B \cdot T = -6173$ ,  $B \cdot R = -3706$  and an error in time of 0.008 days. The entire problem was targeted (to errors of  $B \cdot T = 1000$ ,  $B \cdot R = 200$ ,  $t_{SI} = 0.004$ ) at a final accuracy level of  $5.0 \times 10^{-6}$  in 416 sec including the 220 sec spent in outer targeting.

TABLE 19.- DEMONSTRATION OF BAD-STEP CHECK

Iterate	$V_x$ , km/sec	$V_y$ , km/sec	$V_z$ , km/sec	$B \cdot T$ , km	$B \cdot R$ , km	$t_{SI}$	Error measure
0	1.575340	-20.817502	1.571208	-765 199	-469 501	43.933	1 281 800
<sup>a</sup> 1	1.412716	-20.606402	1.128612	(Missed artificial sphere of influence)			
1	1.534684	-20.764727	1.460559	-682 864	-415 132	44.024	1 135 996
<sup>a</sup> 2	1.397888	-20.597383	1.123011	(Missed artificial sphere of influence)			
2	1.500485	-20.722891	1.376172	-576 213	-345 480	44.134	948 693
3	1.424208	-20.636389	1.215709	-224 998	-130 729	44.702	385 527
<sup>b</sup> 4	1.497700	-20.726740	1.434823	-412 104	-190 780	44.182	625 084
4	1.442581	-20.658977	1.270487	-193 589	-110 660	44.637	327 549
<sup>b</sup> 5	1.504156	-20.736292	1.454002	-282 006	-130 178	44.227	429 884
5	1.457975	-20.678305	1.316366	-160 072	-91 281	44.592	270 153
<sup>b</sup> 6	1.470842	-20.694737	1.355106	-132 022	-74 950	44.556	222 172
7	1.513802	-20.750422	1.484726	-137 682	-63 263	44.319	209,445
8	1.523054	-20.763977	1.513422	-18 519	-11 118	44.412	30 437
Target conditions (outer targeting):				0	0	44.404	
<sup>a</sup> Iterate misses artificial sphere of influence. Velocity correction is reduced by one-quarter to generate new iterate. <sup>b</sup> Iterate increases error of previous iterate. Velocity correction is reduced by one-quarter to generate new iterate.							

5. Other applications.- The targeting program is applicable to many other problems than just the simple probe problems discussed above. One application handled easily by the program involved midcourse corrections. A nominal trajectory was first targeted at the  $2.5 \times 10^{-5}$  accuracy level departing from Earth on 7/24/73 and arriving at Mars on 2/7/74. The sun was the only other gravitational body included in the integration. It was desired to analyze the effect of using midcourse corrections to delay the final arrival at Mars by five days. The times for the corrections were set at five, 25, and 50 days after injection. The position and velocity of the nominal trajectory were recorded at these times and read into the targeting program as injection conditions. The target conditions were identical for each of the three problems ( $ITARG = 5$ ,  $i_{CA} = 38^\circ$ ,  $r_{CA} = 4800$  km,  $t_{CA} = 2/12/74$ ). The differences between these targeted velocities and the velocities at the respective times along the original trajectory then constituted the size of the midcourse correction needed at each candidate time of midcourse correction. The results of this study are summarized in table 20. As can be seen the minimum correction occurs at the third candidate time.

TABLE 20.- ANALYSIS OF MINIMUM VELOCITY MIDCOURSE CORRECTION

Trajectory	$V_x$ , km/sec	$V_y$ , km/sec	$V_z$ , km/sec	B·T, km	B·R, km	$t_{SI}$	$\Delta V$
Nominal at 5 days	26.51898	19.62386	1.2332	4705	6820	63.378	0.13970
Targeted correction	26.49243	19.60173	1.36855	4876	6901	68.315	
Nominal at 25 days	19.28783	26.21346	1.11532	4705	6820	63.378	0.09917
Targeted correction	19.25626	26.19625	1.20774	4892	6931	68.301	
Nominal at 50 days	8.52058	30.02813	.84344	4705	6820	63.378	0.08999
Targeted correction	8.47260	30.01004	.91740	4889	6954	68.290	

A second application of the targeting program has been in the extremely sensitive problem of generating swingby trajectories. A nominal set of target conditions based on a patched conic trajectory originating from Earth and passing near Jupiter, Saturn, and Pluto was used as a profile for the run. The data of that patched conic are summarized below:

Launch date: 8/27/77;

Arrival at Jupiter: + 509 days, B·T = 715 906, B·R = 26 876;

Arrival at Saturn: + 1087 days, B·T = 139 232, B·T = 594 928;

Arrival at Pluto: + 3021 days.

The integration model for the trajectory used an accuracy level of  $1.25 \times 10^{-4}$  with the Sun, Earth, Jupiter, Saturn, and Pluto as gravitational bodies. The trajectory was first targeted to the given patched conic conditions at Jupiter. The injection conditions thus generated yield a trajectory missing the sphere of influence of Saturn. Four iterations of outer targeting yielded a trajectory intersecting the sphere of influence of Saturn. Four iterations of outer targeting yielded a trajectory intersecting the sphere of influence of Saturn. However, six iterations were then required to target to the given target conditions at Saturn. These new injection conditions generate a trajectory that misses Pluto by 121 million km. The entire process could be repeated to target to given conditions at Pluto if desired. The targeting for this swingby mission is summarized in table 21.

Experience has indicated that these swingby trajectories are extremely sensitive problems. Selection of an appropriate velocity increment to use in constructing the state transition matrix is of crucial importance. Undoubtedly major revisions should be made in the targeting scheme before it can be used efficiently for targeting swingby trajectories.

TABLE 21.- TARGETING OF SWINGBY MISSION

	Iterations required	Computer time required, sec	Target injection velocity, km/sec			Target conditions		
			V <sub>x</sub>	V <sub>y</sub>	V <sub>z</sub>	B·T, km	B·R, km	t <sub>SI</sub>
Earth-Jupiter	6	104	19.959364	40.225501	0.630516	715 875	26 876	12/7/78
Earth-Saturn (outer targeting)	4	405	20.082143	40.037109	0.689713	-302 127	-203 541	7/27/80
Earth-Saturn (inner targeting)	6	627	20.079305	40.041451	0.688398	146 456	595 915	7/27/80

### C. Error Analysis Mode

This section demonstrates the uses of the error analysis mode. Many actual runs have been made and the results are indicative of the kind of information that can be obtained from the error analysis mode of STEAP. A basic Mars-Viking run will be analyzed first in detail. Several additional runs will be presented to show how key trajectory parameters vary under differing assumptions. The last set of data contains a midcourse correction schedule analysis for a Venus mission from the proposed Planetary Explorer launch window.

1. Base Mars-Viking run.- Table 22 presents the input used for an example run of the error analysis mode. The nominal trajectory was chosen from the Mars-Viking launch window and was targeted, to satisfy mission constraints, by the targeting mode of STEAP. The launch date for the base run was July 24, 1973, and the time of closest approach to Mars was February 16, 1974. The injection conditions given in the table are consistent with a launch profile that assumes a Cape Kennedy launch and subsequent injection from a nominal, circular parking orbit about the Earth.

The targeting conditions in the Mars impact parameter plane were given by  $B \cdot T = 4938.86$  km,  $B \cdot R = 7179.34$  km, and  $t_{SI} =$  February 13, 19 hr, 14 min, 33.885 sec, 1974. Five bodies were used in the trajectory computation and the integration accuracy level corresponded to a fixed true anomaly increment of 7.696 mrad. No dynamic noise was included in the base run and state transition matrices were computed from the analytical patched conic formulas.

The measurement schedule used for the analysis is given in detail in table 22. All range measurements were assumed to have a variance of  $9 \text{ m}^2$  ( $1\sigma = 3 \text{ m}$ ) and all range rate measurements had variances of  $9 \text{ mm}^2/\text{sec}^2$  ( $1\sigma = 3 \text{ mm/sec}$ ). Three guidance corrections, all using the 3-variable B-plane policy, were made at 5, 30, and 197 days after injection. Execution error variances at the times of midcourse correction are included in the table and they correspond to  $1\sigma$  errors of 0.002 for the proportionality error, 2 cm/sec for the resolution error, and 7.1 mrad for each of the pointing errors.

TABLE 22.- CONDITIONS FOR BASE ERROR ANALYSIS RUN

Injection date: July 24, 9 hr, 25 min, 47.639 sec, 1973

Closest approach date: Feb. 16, 0 hr, 1 min, 49.377 sec, 1974

Injection conditions: (geocentric ecliptic coordinates)

X = -1090.0529 km

$\dot{X}$  = 9.401527 km/sec

Y = -6550.0511 km

$\dot{Y}$  = -2.876404 km/sec

Z = 75.6015 km

$\dot{Z}$  = 6.200795 km/sec

$r_E$  = 6640.5648 km

$V_E$  = 11.623781 km/sec

Mars sphere of influence conditions:

B = 8714.08 km

B · R = 7179.34 km

B · T = 4938.86 km

$t_{SI}$  = Feb. 13, 19 hr, 14 min, 33.885 sec, 1974

Ephemeris included Earth, Mars, Sun, Jupiter, Moon

Accuracy level:  $5 \times 10^{-6}$

True anomaly: 7.696 mrad

No dynamic noise

Star transition matrices from patched conic

Measurement schedule - Goldstone measured range and range rate 5 times at 0.2, 1.2, 2.2, 3.2, 4.2  
 Goldstone measured range rate 19 times at 5.3, 9.3, 13.3, 17.3, 21.3, 25.3,  
 29.3, 34.1, 64.1, 94.1, 124.1, 154.1, 184.1, 187.3, 190.3, 193.3, 196.3,  
 199.3, 202.3  
 Madrid measured range and range rate 12 times at 0.6, 1.6, 2.6, 3.6, 4.6,  
 6.7, 10.7, 14.7, 18.7, 22.7, 26.7, 30.7  
 Madrid measured range rate 11 times at 44.5, 74.5, 104.5, 134.5, 164.5, 194.5,  
 203.5, 204.5, 205.5, 206.5, 207.5  
 Canberra measured range and range rate 12 times at 0.9, 1.9, 2.9, 3.9, 4.9,  
 8.0, 12.0, 16.0, 20.0, 24.0, 28.0, 32.0  
 Canberra measured range rate 6 times at 54.8, 84.8, 114.8, 144.8, 174.8,  
 204.8

Total of 65 measurement times

Measurement accuracies:  $\sigma_D^2 = 9 \text{ m}^2$  for all ranges

$\sigma_{\dot{D}}^2 = 9 \text{ mm}^2/\text{sec}^2$  for all range rates

Injection covariance diagonal:  $\sigma_P^2 = 1 \text{ km}^2$  for all positions

$\sigma_V^2 = 9 \text{ m}^2/\text{sec}^2$  for all velocities

Guidance corrections at 5, 30, 197 days; three variable B-plane policy

Execution errors:  $\sigma_{pro}^2 = 4 \times 10^{-6}$

$\sigma_{res}^2 = 4 \text{ cm}^2/\text{sec}^2$

$\sigma_{\delta\alpha}^2 = \sigma_{\delta\beta}^2 = 5 \times 10^{-5} \text{ rad}^2$

Recall that there are three major outputs of the error analysis mode. The first of these is the navigation uncertainty covariance matrix at selected epochs along the orbit. A second is target condition variations about the nominal, which indicate how closely the corrected trajectory is likely to meet the specified target conditions. The third significant output is related to the likely size of the midcourse corrections and is important for fuel loading purposes.

Table 23 presents a detailed tabulation of the diagonal elements of the state vector associated covariance matrices for the base run. The symbols + and - in the trajectory time column indicate values before and after processing the measurement made at the specified time. The measurement kind column refers to the type of measurement being processed; thus at 0.9 days after injection a measurement of range and range rate ( $D_3$  and  $\dot{D}_3$ ) was taken from Canberra. The initial position variances were all  $1 \text{ km}^2$  and the initial velocity variances, also uncorrelated, were assumed to be  $9 \text{ m}^2/\text{sec}^2$ . The symbols  $G_1$ ,  $G_2$ , and  $G_3$  in the measurement number column indicate the times of the three guidance corrections.

The position uncertainties displayed in the table, two of which are plotted in standard deviation form ( $3\sigma$ ) in figure 11, are fairly typical of similar runs. Due to the sensitivity of the outgoing hyperbola, which produces large state transition matrices, the position uncertainty variances increase initially. After one day and three measurements of range and range rate, the position navigation uncertainties have approximately returned to their initial values. Subsequent tracking and data processing reduces the position errors to about  $0.75 \text{ km}$  ( $3\sigma$ ) after three days and the variances stay more or less the same until the first guidance correction five days after injection. It is interesting to note from table 23 the affect of each individual measurement on the processed covariance matrices. The sixth measurement, for example, which was a range and range rate measurement from Canberra, only reduced  $\sigma_z^2$  from 0.531 to 0.423. By contrast, the seventh measurement, range and range rate from Goldstone, reduced the same quantity by more than one order of magnitude. The geometry of the tracking station-trajectory relationship plus other factors related to previous measurements doubtless explain why one measurement reduces uncertainties more than another.



TABLE 23.- NAVIGATION UNCERTAINTIES FOR BASE ERROR ANALYSIS RUN

Data Trajectory time, days	Meas No.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
0	0	----	1.000	1.000	1.000	9.000	9.000	9.000
0.2 <sup>-</sup>	0		1.035 x 10 <sup>4</sup>	2.410 x 10 <sup>3</sup>	4.885 x 10 <sup>3</sup>	5.115 x 10 <sup>1</sup>	4.199	2.177 x 10 <sup>1</sup>
0.2 <sup>+</sup>	1	D <sub>1</sub> , D <sub>1</sub>	2.130 x 10 <sup>2</sup>	1.626 x 10 <sup>3</sup>	1.856 x 10 <sup>2</sup>	7.575 x 10 <sup>-1</sup>	3.504	5.639 x 10 <sup>-1</sup>
0.6 <sup>-</sup>	1		1.804 x 10 <sup>3</sup>	1.003 x 10 <sup>4</sup>	1.409 x 10 <sup>3</sup>	5.967 x 10 <sup>-1</sup>	2.755	4.420 x 10 <sup>-1</sup>
0.6 <sup>+</sup>	2	D <sub>2</sub> , D <sub>2</sub>	2.014 x 10 <sup>1</sup>	2.753 x 10 <sup>1</sup>	4.618	1.063 x 10 <sup>-2</sup>	6.948 x 10 <sup>-3</sup>	4.943 x 10 <sup>-3</sup>
0.9 <sup>-</sup>	2		5.055 x 10 <sup>1</sup>	5.415 x 10 <sup>1</sup>	1.522 x 10 <sup>1</sup>	1.022 x 10 <sup>-2</sup>	6.554 x 10 <sup>-3</sup>	4.837 x 10 <sup>-3</sup>
0.9 <sup>+</sup>	3	D <sub>3</sub> , D <sub>3</sub>	1.586	8.322 x 10 <sup>1</sup>	3.220	3.631 x 10 <sup>-4</sup>	2.113 x 10 <sup>-4</sup>	7.289 x 10 <sup>-4</sup>
1.2 <sup>-</sup>	3		2.884	1.562	6.025	3.551 x 10 <sup>-4</sup>	2.073 x 10 <sup>-4</sup>	7.138 x 10 <sup>-4</sup>
1.2 <sup>+</sup>	4	D <sub>1</sub> , D <sub>1</sub>	6.380 x 10 <sup>-2</sup>	9.757 x 10 <sup>-1</sup>	7.406 x 10 <sup>-1</sup>	6.743 x 10 <sup>-5</sup>	1.331 x 10 <sup>-4</sup>	1.846 x 10 <sup>-4</sup>
1.6 <sup>-</sup>	4		2.144 x 10 <sup>-1</sup>	1.828	1.667	6.706 x 10 <sup>-5</sup>	1.309 x 10 <sup>-4</sup>	1.822 x 10 <sup>-4</sup>
1.6 <sup>+</sup>	5	D <sub>2</sub> , D <sub>2</sub>	1.533 x 10 <sup>-1</sup>	7.853 x 10 <sup>-2</sup>	2.556 x 10 <sup>-1</sup>	6.421 x 10 <sup>-5</sup>	2.617 x 10 <sup>-5</sup>	7.670 x 10 <sup>-5</sup>
1.9 <sup>-</sup>	5		3.499 x 10 <sup>-1</sup>	1.495 x 10 <sup>-1</sup>	5.310 x 10 <sup>-1</sup>	6.394 x 10 <sup>-5</sup>	2.608 x 10 <sup>-5</sup>	7.633 x 10 <sup>-5</sup>
1.9 <sup>+</sup>	6	D <sub>3</sub> , D <sub>3</sub>	2.399 x 10 <sup>-1</sup>	1.031 x 10 <sup>-1</sup>	4.227 x 10 <sup>-1</sup>	5.024 x 10 <sup>-5</sup>	2.038 x 10 <sup>-5</sup>	6.248 x 10 <sup>-5</sup>
2.2 <sup>-</sup>	6		4.492 x 10 <sup>-1</sup>	1.784 x 10 <sup>-1</sup>	7.272 x 10 <sup>-1</sup>	5.005 x 10 <sup>-5</sup>	2.032 x 10 <sup>-5</sup>	6.221 x 10 <sup>-5</sup>
2.2 <sup>+</sup>	7	D <sub>1</sub> , D <sub>1</sub>	4.624 x 10 <sup>-2</sup>	9.503 x 10 <sup>-2</sup>	5.225 x 10 <sup>-2</sup>	3.252 x 10 <sup>-6</sup>	3.451 x 10 <sup>-6</sup>	3.410 x 10 <sup>-6</sup>
2.6 <sup>-</sup>	7		7.354 x 10 <sup>-2</sup>	1.357 x 10 <sup>-1</sup>	8.216 x 10 <sup>-2</sup>	3.236 x 10 <sup>-6</sup>	3.428 x 10 <sup>-6</sup>	3.394 x 10 <sup>-6</sup>
2.6 <sup>+</sup>	8	D <sub>2</sub> , D <sub>2</sub>	5.965 x 10 <sup>-2</sup>	8.581 x 10 <sup>-2</sup>	7.701 x 10 <sup>-2</sup>	2.600 x 10 <sup>-6</sup>	1.982 x 10 <sup>-6</sup>	3.382 x 10 <sup>-6</sup>
2.9 <sup>-</sup>	9	D <sub>3</sub> , D <sub>3</sub>	6.695 x 10 <sup>-2</sup>	9.505 x 10 <sup>-2</sup>	1.005 x 10 <sup>-1</sup>	2.398 x 10 <sup>-6</sup>	1.838 x 10 <sup>-6</sup>	3.297 x 10 <sup>-6</sup>
3.2 <sup>+</sup>	10	D <sub>1</sub> , D <sub>1</sub>	5.931 x 10 <sup>-2</sup>	1.143 x 10 <sup>-1</sup>	6.436 x 10 <sup>-2</sup>	1.305 x 10 <sup>-6</sup>	1.608 x 10 <sup>-6</sup>	1.453 x 10 <sup>-6</sup>
3.6 <sup>+</sup>	11	D <sub>2</sub> , D <sub>2</sub>	6.854 x 10 <sup>-2</sup>	1.106 x 10 <sup>-1</sup>	8.265 x 10 <sup>-2</sup>	1.124 x 10 <sup>-6</sup>	1.166 x 10 <sup>-6</sup>	1.443 x 10 <sup>-6</sup>
3.9 <sup>+</sup>	12	D <sub>3</sub> , D <sub>3</sub>	7.242 x 10 <sup>-2</sup>	1.196 x 10 <sup>-1</sup>	9.802 x 10 <sup>-2</sup>	1.041 x 10 <sup>-6</sup>	1.102 x 10 <sup>-6</sup>	1.415 x 10 <sup>-6</sup>
4.2 <sup>+</sup>	13	D <sub>1</sub> , D <sub>1</sub>	7.276 x 10 <sup>-2</sup>	1.383 x 10 <sup>-1</sup>	7.722 x 10 <sup>-2</sup>	7.847 x 10 <sup>-7</sup>	1.072 x 10 <sup>-6</sup>	8.780 x 10 <sup>-7</sup>
4.6 <sup>+</sup>	14	D <sub>2</sub> , D <sub>2</sub>	8.047 x 10 <sup>-2</sup>	1.365 x 10 <sup>-1</sup>	9.274 x 10 <sup>-2</sup>	7.006 x 10 <sup>-7</sup>	8.572 x 10 <sup>-7</sup>	8.698 x 10 <sup>-7</sup>
4.9 <sup>+</sup>	15	D <sub>3</sub> , D <sub>3</sub>	8.336 x 10 <sup>-2</sup>	1.457 x 10 <sup>-1</sup>	1.051 x 10 <sup>-2</sup>	6.531 x 10 <sup>-7</sup>	8.189 x 10 <sup>-7</sup>	8.546 x 10 <sup>-7</sup>
5.0 <sup>-</sup>	G <sub>1</sub>	----	8.732 x 10 <sup>-2</sup>	1.516 x 10 <sup>-1</sup>	1.102 x 10 <sup>-1</sup>	6.533 x 10 <sup>-7</sup>	8.193 x 10 <sup>-7</sup>	8.544 x 10 <sup>-7</sup>
5.0 <sup>+</sup>						1.311 x 10 <sup>-3</sup>	3.727 x 10 <sup>-3</sup>	3.124 x 10 <sup>-3</sup>
5.3 <sup>-</sup>	15	D <sub>1</sub>	9.744 x 10 <sup>-1</sup>	2.656	2.210	1.311 x 10 <sup>-3</sup>	3.727 x 10 <sup>-3</sup>	3.124 x 10 <sup>-3</sup>
5.3 <sup>+</sup>	16		4.788 x 10 <sup>-1</sup>	1.738	1.239	5.695 x 10 <sup>-4</sup>	2.349 x 10 <sup>-3</sup>	1.669 x 10 <sup>-3</sup>
6.7 <sup>-</sup>	16	D <sub>2</sub> , D <sub>2</sub>	1.243 x 10 <sup>1</sup>	5.089 x 10 <sup>1</sup>	3.615 x 10 <sup>1</sup>	5.706 x 10 <sup>-4</sup>	2.352 x 10 <sup>-3</sup>	1.668 x 10 <sup>-3</sup>
6.7 <sup>+</sup>	17	D <sub>2</sub> , D <sub>2</sub>	1.203 x 10 <sup>1</sup>	4.386 x 10 <sup>1</sup>	3.211 x 10 <sup>1</sup>	5.521 x 10 <sup>-4</sup>	2.028 x 10 <sup>-3</sup>	1.481 x 10 <sup>-3</sup>
8.0 <sup>-</sup>	17	D <sub>3</sub> , D <sub>3</sub>	3.727 x 10 <sup>1</sup>	1.363 x 10 <sup>2</sup>	9.968 x 10 <sup>1</sup>	5.545 x 10 <sup>-4</sup>	2.033 x 10 <sup>-3</sup>	1.479 x 10 <sup>-3</sup>
8.0 <sup>+</sup>	18		2.942 x 10 <sup>1</sup>	9.008 x 10 <sup>1</sup>	1.650 x 10 <sup>1</sup>	4.393 x 10 <sup>-4</sup>	1.338 x 10 <sup>-3</sup>	2.471 x 10 <sup>-4</sup>
9.3 <sup>+</sup>	19	D <sub>1</sub> , D <sub>1</sub>	5.553 x 10 <sup>1</sup>	1.699 x 10 <sup>2</sup>	3.252 x 10 <sup>1</sup>	4.062 x 10 <sup>-4</sup>	1.234 x 10 <sup>-3</sup>	2.364 x 10 <sup>-4</sup>
10.7 <sup>+</sup>	20	D <sub>2</sub> , D <sub>2</sub>	3.232	1.917 x 10 <sup>1</sup>	3.227 x 10 <sup>1</sup>	1.295 x 10 <sup>-5</sup>	7.452 x 10 <sup>-5</sup>	1.332 x 10 <sup>-4</sup>
12.0 <sup>+</sup>	21	D <sub>3</sub> , D <sub>3</sub>	4.079	1.922 x 10 <sup>1</sup>	1.558 x 10 <sup>1</sup>	1.098 x 10 <sup>-5</sup>	4.939 x 10 <sup>-5</sup>	4.222 x 10 <sup>-5</sup>
13.3 <sup>+</sup>	22	D <sub>1</sub>	5.685	2.664 x 10 <sup>1</sup>	2.183 x 10 <sup>1</sup>	1.111 x 10 <sup>-5</sup>	4.948 x 10 <sup>-5</sup>	4.195 x 10 <sup>-5</sup>

TABLE 23.- NAVIGATION UNCERTAINTIES FOR BASE ERROR ANALYSIS RUN - Concluded

Data Trajectory time, days	Meas No.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
14.7 <sup>+</sup>	23	$\dot{D}_2, D_2$	6.571	$1.995 \times 10^1$	$1.726 \times 10^1$	$9.869 \times 10^{-6}$	$2.872 \times 10^{-5}$	$2.450 \times 10^{-5}$
16.0 <sup>+</sup>	24	$\dot{D}_3, D_3$	1.779	$1.511 \times 10^1$	$2.202 \times 10^1$	$2.318 \times 10^{-6}$	$1.713 \times 10^{-5}$	$2.414 \times 10^{-5}$
20.0 <sup>+</sup>	27	$\dot{D}_3, D_3$	$9.600 \times 10^{-1}$	2.608	$1.304 \times 10^1$	$6.712 \times 10^{-7}$	$1.695 \times 10^{-6}$	$7.609 \times 10^{-6}$
24.0 <sup>+</sup>	30	$\dot{D}_3, D_3$	$9.385 \times 10^{-1}$	1.659	$1.209 \times 10^1$	$3.817 \times 10^{-7}$	$6.418 \times 10^{-7}$	$4.294 \times 10^{-6}$
28.0 <sup>+</sup>	33	$\dot{D}_3, D_3$	$8.737 \times 10^{-1}$	1.529	$1.167 \times 10^1$	$2.264 \times 10^{-7}$	$3.839 \times 10^{-7}$	$2.743 \times 10^{-6}$
29.3 <sup>+</sup>	34	$\dot{D}_1$	$9.733 \times 10^{-1}$	1.701	$1.295 \times 10^1$	$2.249 \times 10^{-7}$	$3.771 \times 10^{-7}$	$2.696 \times 10^{-6}$
30.0 <sup>-</sup>	G <sub>2</sub>	----	1.029	1.797	$1.368 \times 10^1$	$2.242 \times 10^{-7}$	$3.732 \times 10^{-7}$	$2.670 \times 10^{-6}$
30.0 <sup>+</sup>						$2.747 \times 10^{-4}$	$4.450 \times 10^{-5}$	$8.468 \times 10^{-5}$
30.7 <sup>+</sup>	35	$\dot{D}_2, D_2$	1.086	1.896	$1.441 \times 10^1$	$4.746 \times 10^{-7}$	$6.183 \times 10^{-7}$	$3.337 \times 10^{-6}$
32.0 <sup>+</sup>	36	$D_3, \dot{D}_3$	1.109	1.698	$1.386 \times 10^1$	$4.240 \times 10^{-7}$	$5.104 \times 10^{-7}$	$2.873 \times 10^{-6}$
34.1 <sup>+</sup>	37	$\dot{D}_1$	1.296	1.970	$1.597 \times 10^1$	$4.255 \times 10^{-7}$	$4.997 \times 10^{-7}$	$2.798 \times 10^{-6}$
64.1 <sup>+</sup>	40	$\dot{D}_1$	7.584	8.034	$5.489 \times 10^1$	$6.877 \times 10^{-7}$	$3.045 \times 10^{-7}$	$1.426 \times 10^{-6}$
94.1 <sup>+</sup>	43	$\dot{D}_1$	$1.891 \times 10^1$	$1.501 \times 10^1$	$6.561 \times 10^1$	$9.946 \times 10^{-7}$	$1.864 \times 10^{-7}$	$3.177 \times 10^{-7}$
124.1 <sup>+</sup>	46	$\dot{D}_1$	$3.179 \times 10^1$	$2.041 \times 10^1$	$4.370 \times 10^1$	$9.274 \times 10^{-7}$	$2.852 \times 10^{-7}$	$7.270 \times 10^{-8}$
154.1 <sup>+</sup>	49	$\dot{D}_1$	$4.762 \times 10^1$	$2.471 \times 10^1$	$2.381 \times 10^1$	$6.620 \times 10^{-7}$	$5.388 \times 10^{-7}$	$9.391 \times 10^{-8}$
184.1 <sup>+</sup>	52	$\dot{D}_1$	$6.474 \times 10^1$	$3.106 \times 10^1$	$1.330 \times 10^1$	$3.628 \times 10^{-7}$	$8.445 \times 10^{-7}$	$1.160 \times 10^{-7}$
194.5 <sup>+</sup>	56	$\dot{D}_2$	$6.687 \times 10^1$	$2.494 \times 10^1$	9.694	$2.161 \times 10^{-7}$	$7.347 \times 10^{-7}$	$9.824 \times 10^{-8}$
197.0 <sup>-</sup>	G <sub>3</sub>	----	$6.764 \times 10^1$	$2.422 \times 10^1$	9.231	$1.916 \times 10^{-7}$	$7.226 \times 10^{-7}$	$9.543 \times 10^{-8}$
197.0 <sup>+</sup>						$2.495 \times 10^{-4}$	$8.849 \times 10^{-5}$	$1.206 \times 10^{-4}$
199.3 <sup>+</sup>	58	$\dot{D}_1$	$7.352 \times 10^1$	$2.897 \times 10^1$	$1.329 \times 10^1$	$1.502 \times 10^{-4}$	$8.802 \times 10^{-5}$	$9.572 \times 10^{-5}$
202.3 <sup>+</sup>	59	$\dot{D}_1$	$9.893 \times 10^1$	$4.504 \times 10^1$	$2.998 \times 10^1$	$1.496 \times 10^{-4}$	$8.531 \times 10^{-5}$	$9.551 \times 10^{-5}$
<sup>a</sup> 203.5 <sup>-</sup>	59	$\dot{D}_2$	$1.147 \times 10^2$	$5.488 \times 10^1$	$4.024 \times 10^1$	$1.495 \times 10^{-4}$	$8.552 \times 10^{-5}$	$9.541 \times 10^{-5}$
203.5 <sup>+</sup>	60		$1.146 \times 10^2$	$5.322 \times 10^1$	$3.997 \times 10^1$	$1.489 \times 10^{-4}$	$8.185 \times 10^{-5}$	$9.478 \times 10^{-5}$
<sup>b</sup> 204.5 <sup>-</sup>	60	$\dot{D}_2$	$1.301 \times 10^2$	$6.238 \times 10^1$	$5.000 \times 10^1$	$1.487 \times 10^{-4}$	$8.204 \times 10^{-5}$	$9.467 \times 10^{-5}$
204.5 <sup>+</sup>	61		$1.292 \times 10^2$	$5.997 \times 10^1$	$4.927 \times 10^1$	$1.463 \times 10^{-4}$	$7.753 \times 10^{-5}$	$9.317 \times 10^{-5}$
<sup>c</sup> 205.5 <sup>-</sup>	62	$\dot{D}_2$	$1.459 \times 10^2$	$6.703 \times 10^1$	$5.940 \times 10^1$	$1.313 \times 10^{-4}$	$6.948 \times 10^{-5}$	$8.637 \times 10^{-5}$
205.5 <sup>+</sup>	63		$1.457 \times 10^2$	$6.095 \times 10^1$	$5.505 \times 10^1$	$1.285 \times 10^{-4}$	$6.421 \times 10^{-5}$	$8.231 \times 10^{-5}$
<sup>d</sup> 206.5 <sup>-</sup>	63	$\dot{D}_2$	$1.619 \times 10^2$	$6.952 \times 10^1$	$6.362 \times 10^1$	$6.377 \times 10^{-3}$	$8.296 \times 10^{-3}$	$2.630 \times 10^{-3}$
206.5 <sup>+</sup>	64		$1.078 \times 10^2$	$5.073 \times 10^1$	$4.567 \times 10^1$	$2.393 \times 10^{-3}$	$1.480 \times 10^{-3}$	$1.452 \times 10^{-3}$
<sup>e</sup> 207.5 <sup>-</sup>	64	$\dot{D}_2$	$6.081 \times 10^4$	$2.532 \times 10^4$	$4.378 \times 10^4$	9.772	4.128	6.736
207.5 <sup>+</sup>	65		$4.577 \times 10^4$	$2.447 \times 10^4$	$4.214 \times 10^4$	7.422	4.040	6.563

<sup>a</sup>Distance from Mars at 203.5 = 793 000 km.  
<sup>b</sup>Distance from Mars at 204.5 = 543 000 km.  
<sup>c</sup>Distance from Mars at 205.5 = 291 000 km.  
<sup>d</sup>Distance from Mars at 206.5 = 33 000 km.  
<sup>e</sup>Distance from Mars at 207.5 = 236 000 km.

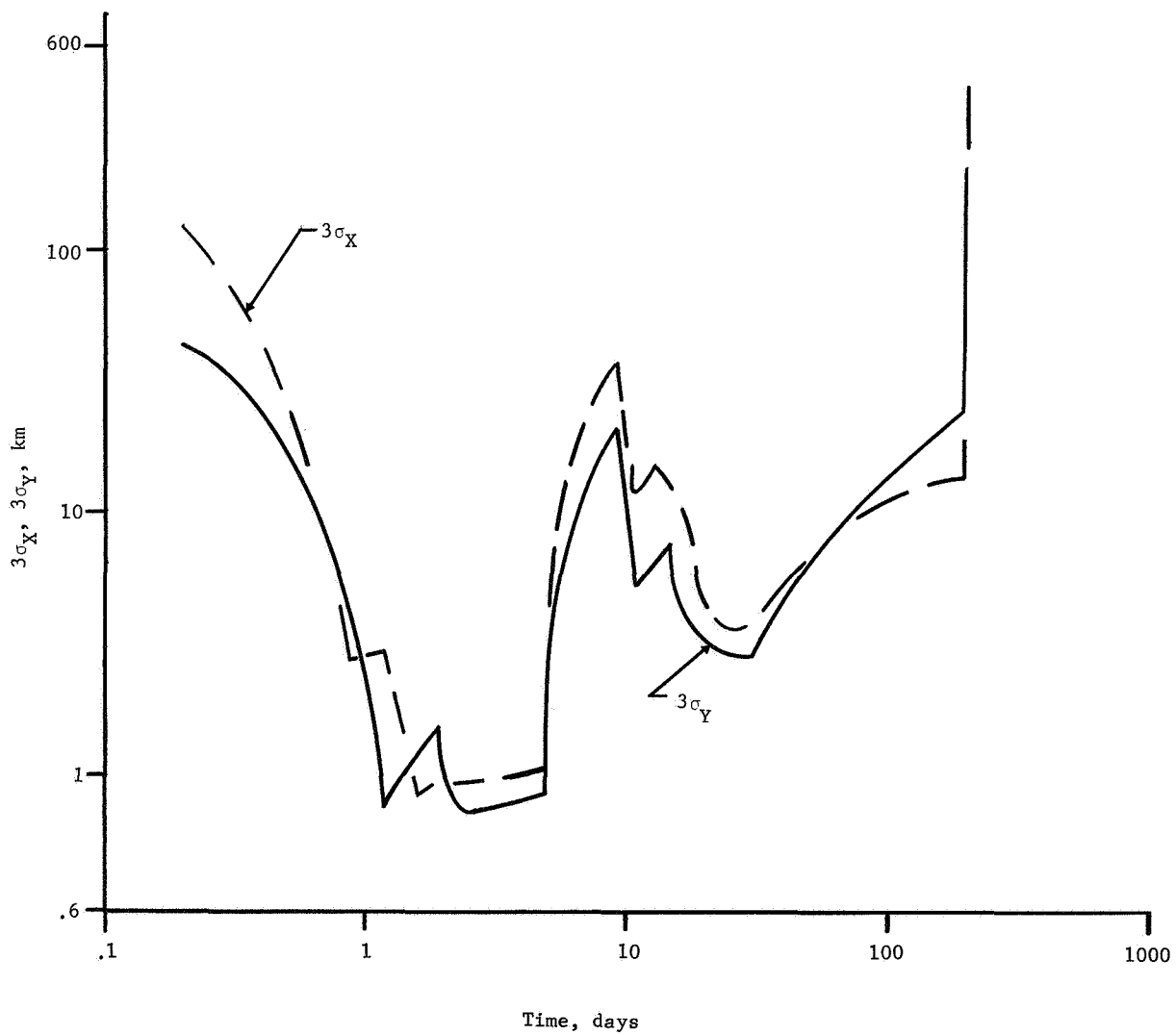


Figure 11.- Processed Navigation Uncertainties from Base Error Analysis Run

At the time of the first guidance correction  $(G_1)$ , the  $3\sigma$  position uncertainties are approximately 0.87 km, 1.17 km, and 0.98 km, respectively. Although the execution error covariance matrix  $\tilde{Q}$  adds immediately only to the velocity uncertainties, the position errors increase briefly while the execution errors are being filtered by the algorithm. After processing the range and range rate measurements from Goldstone 9.3 days after injection, the  $3\sigma$  position uncertainties have reached magnitudes of 22.5 km, 39 km, and 16.8 km. The X and Y navigation errors then drop rapidly and level off around a  $3\sigma$  value of 3 km prior to the second midcourse correction. The uncertainty in the Z component of position is not significantly reduced during this same period and is still 11.1 km ( $3\sigma$ ) at the time of the second correction. As was mentioned earlier, tracking station-orbit geometry probably accounts for some uncertainties being reduced more than others by the tracking.

After the second guidance correction, several factors contribute to the rising navigation uncertainties. First, execution errors are made at the second correction and are naturally quickly reflected in the navigation uncertainty covariance matrix. Second, tracking measurements are taken at wider intervals and with less frequency. Third, the most important, only range rate measurements are made after the first two observations following the second midcourse correction.

The pattern displayed by the position variances between the second and third midcourse corrections might be called barely controlled. Each measurement more or less counteracts the uncertainty increase between observations. At the time of the third midcourse correction, which is roughly ten days before closest approach to Mars, the position uncertainties are ( $3\sigma$ ) 24.7 km, 14.6 km, and 9.1 km. After the final correction in the example error analysis run, as the hypothetical spacecraft nears Mars, measurements are taken more frequently. The position uncertainties first increase due to the execution errors from the third correction. However, as periapsis is neared, the position uncertainties decrease, although not markedly, because the influence of the third correction execution errors has not been totally eliminated. The position uncertainties at periapsis are shown in table 24 and their  $3\sigma$  levels of 17.504, 18.836, and 13.190 km are less than those at 206.5 days shown on table 23.

TABLE 24.- NAVIGATION UNCERTAINTIES AT CLOSEST APPROACH FOR BASE RUN

Closest approach conditions (in areocentric coordinates)	Covariance matrix information	
$X = 2010.7 \text{ km}$	$\sigma_X^2 = 34.0458 \text{ km}^2$	$\rho_{XY} = -0.39337$
$Y = -1576.43 \text{ km}$	$3\sigma_X = 17.504 \text{ km}$	$\rho_{XZ} = 0.83573$
$Z = -4302.37 \text{ km}$	$\sigma_Y^2 = 39.4216 \text{ km}^2$	$\rho_{YZ} = 0.16975$
$\dot{X} = 2.79597 \text{ km/sec}$	$3\sigma_Y = 18.836 \text{ km}$	$\rho_{\dot{X}\dot{Y}} = -0.71155$
$\dot{Y} = 4.18806$	$\sigma_Z^2 = 19.3314 \text{ km}^2$	$\rho_{\dot{X}\dot{Z}} = 0.72032$
$\dot{Z} = -0.24596$	$3\sigma_Z = 13.190 \text{ km}$	$\rho_{\dot{Y}\dot{Z}} = -0.26447$
$R_p = 5003.87$	$\sigma_{\dot{X}}^2 = 10.2129 \text{ m}^2/\text{sec}^2$	$\rho_{\dot{X}\dot{X}} = -0.98982$
$V_p = 5.04160$	$3\sigma_{\dot{X}} = 9.587 \text{ m/sec}$	$\rho_{\dot{Y}\dot{Y}} = 0.22906$
	$\sigma_{\dot{Y}}^2 = 6.4610 \text{ m}^2/\text{sec}^2$	$\rho_{\dot{Z}\dot{Z}} = 0.86551$
	$3\sigma_{\dot{Y}} = 7.625 \text{ m/sec}$	$\rho_{\dot{Y}\dot{Z}} = 0.96696$
	$\sigma_{\dot{Z}}^2 = 8.6935^2 \text{ m}^2/\text{sec}^2$	$\rho_{\dot{Z}\dot{Y}} = 0.99791$
	$3\sigma_{\dot{Z}} = 8.846 \text{ m/sec}$	

After Mars periapsis passage, the position uncertainties rise at a fantastic rate. This phenomenon was noticed in all the test runs. Coming into the target planet on a hyperbola, the position uncertainties decreased as the trajectory came closer and closer to periapsis. Physically this means that the target planet is influencing families of neighboring trajectories in much the same way and, to continue the analogy, is pulling them together during the approach period. Conversely, after periapsis passage, families of neighboring trajectories are scattered wildly and slight uncertainties at closest approach become quickly magnified. One can immediately conjecture that a similar analysis of a multi-planet swingby mission would demonstrate a considerable loss of navigation accuracy after passing the intermediate planet.

Table 23 also contains a chart of the velocity navigation uncertainties for the same example run. Figure 12 shows the general behavior of the  $\dot{\mathbf{X}}$  velocity uncertainties throughout the example mission. Unlike the position uncertainties, the velocity errors show no initial precipitous rise on the outgoing Earth hyperbola. After one complete cycle of measurements, at 0.9 days after injection, the initial velocity errors of 9 m/sec ( $3\sigma$ ) in all components have been reduced by the navigation process to 0.057, 0.043, 0.069 m/sec, again using  $3\sigma$  numbers. Subsequent measurements before the first midcourse correction regularly reduce the velocity uncertainties, which are not growing measurably between observations, until, at the time of the first correction, their  $3\sigma$  levels are 2.43, 2.71, and 2.76 mm/sec. The spacecraft velocities for the hypothetical example run are very well known at the time of the first correction.

A midcourse correction introduces a discontinuity into the navigation velocity uncertainties. Impulsive velocity maneuvers are assumed by the process, and thus the execution error matrix  $\tilde{\mathbf{Q}}$  discussed earlier is added to the earlier velocity uncertainties at the specific time. Primarily because of the execution errors, the  $3\sigma$  velocity uncertainties after the first midcourse correction have risen to 108, 186, and 168 mm/sec. Figure 12 shows the discontinuities in the velocity uncertainties at the times of midcourse correction.

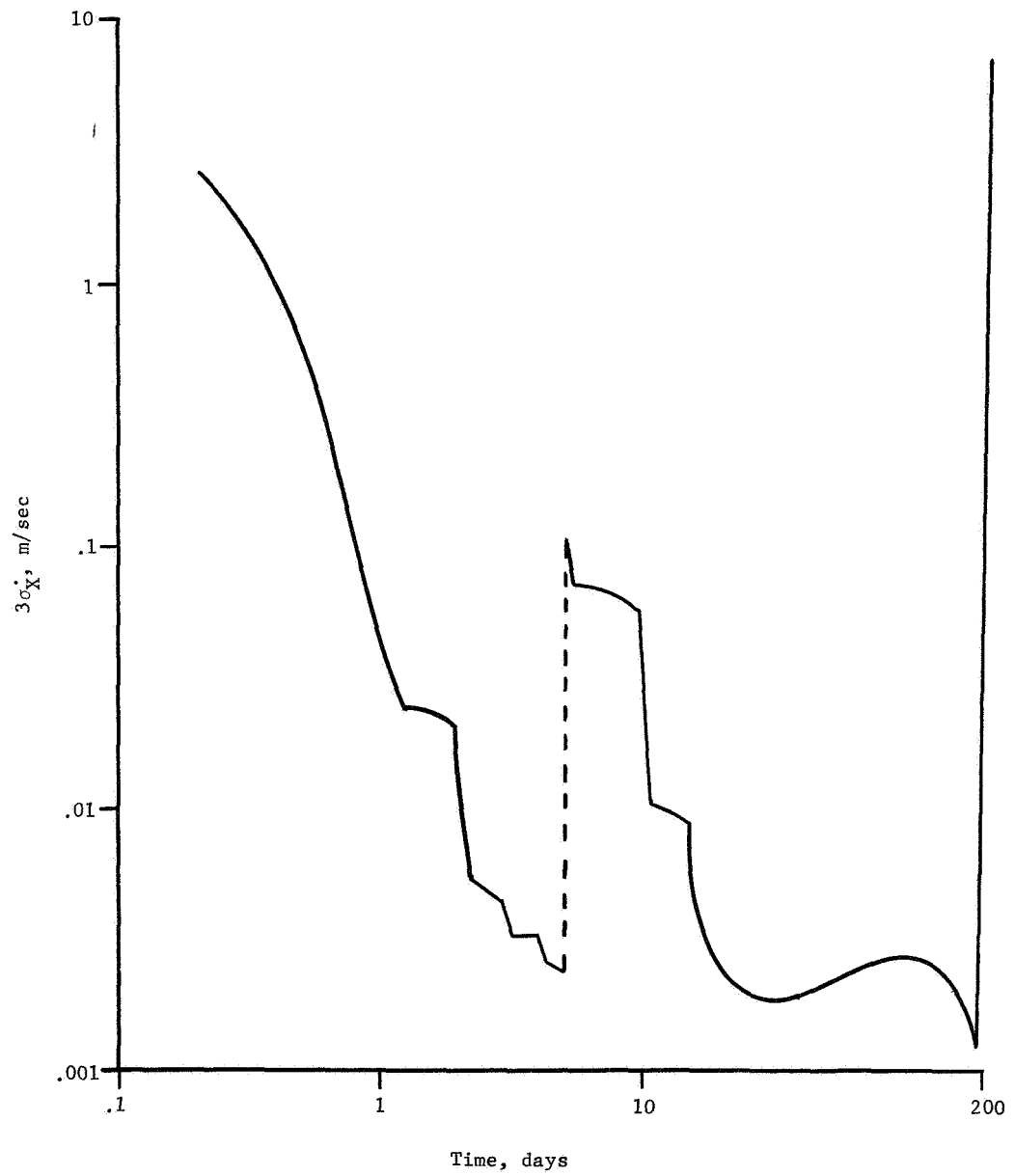


Figure 12.- Navigation Uncertainties from Base Error Analysis Run

Again the tracking-navigation process reduces the velocity uncertainties in a regular fashion before the second assumed midcourse correction 30 days after injection. However, just as the  $Z$  component of position uncertainty failed to come down as much as the other position uncertainties (probably because of the geometrical relationship between the tracking stations and the trajectory over the interval between corrections), so the  $\dot{Z}$  velocity uncertainty is not as small as the other two at the time of the second midcourse correction. Before the second correction, the  $3\sigma$  velocity errors are given by 1.38, 1.83, and 15.6 mm/sec.

Over the long heliocentric phase of the trajectory, before the final midcourse correction, the navigation algorithm filters out the second maneuver execution errors despite the infrequency of the measurements. At the time of the third correction, the  $3\sigma$  levels of velocity uncertainty are down to 1.32, 2.55, and 0.93 mm/sec. Notice that now the  $\dot{Z}$  component is best known, and that the error in the  $\dot{Y}$  component is the largest of the three. The execution errors at the third correction are never reduced considerably during the approach to Mars. This is because of the rapidly increasing velocity of the vehicle as it draws near to the target planet and the concomitant magnification, in terms of errors, of the velocity uncertainties. At closest approach (see table 24), the velocity uncertainties have risen to  $3\sigma$  levels of 9.587, 7.625, and 8.846 m/sec, far above their values during most of the trajectory.

It should be stressed again that all test runs demonstrated the same navigation uncertainty behavior upon passing close to the target planet. During the approach, the position uncertainties decreased and the velocity uncertainties increased rapidly as the planet speeded up the incoming vehicle. Just after periapsis, the position uncertainties were considerably magnified and the velocity errors diminished slowly with the processing of the tracking data. From the point of view of an assumed orbit insertion maneuver at periapsis, it is apparent that the position of the spacecraft at the time of insertion should be well known, but that considerable errors may be present in the estimated velocity.



Table 24 also presents the closest approach conditions for the nominal trajectory of the example run and several of the key correlation coefficients relating uncertainties in the various elements of the state vector. The periapsis height above the assumed spherical Martian surface was 1610.47 km. Three of the correlation coefficients were large enough to create some concern about observability. However, experience indicates that until one of the  $\rho$  values reaches at least 0.999 the estimation algorithm can operate without undue difficulty.

Table 25 gives the  $\Delta V$  information for the base error analysis run. The most likely magnitude of the first correction, whose computation method was discussed in an earlier section of this analytic manual, is 8.638 m/sec. The standard deviation  $\sigma$  about this mean value was calculated as 6.287 m/sec. The underlying probability distribution is not normal or Gaussian, because clearly no negative values can be permitted and the mean minus  $2\sigma$  would give a negative midcourse velocity. A more detailed description of the probabilistic interpretation of these two numbers is given by Hoffman and Young in reference 12.

Recall that the most likely magnitude for the midcourse corrections are computed from navigation uncertainties after the last correction. The state transition matrices are computed between corrections, as well as the guidance matrix  $\Gamma$  that is computed from the specified guidance law. What the value 8.638 m/sec represents, therefore, is a most likely correction that would result if the set of all trajectories whose errors at injection are given by  $P_0$  were actually flown in a Monte Carlo sense.

Given that a correction of some magnitude takes place, one can determine from the same information (the injection covariance, the state transition matrix from injection to the first midcourse, and the guidance matrix) the most likely direction of the correction. The effective most likely correction, given by " $E[\Delta V]$ ", is given in table 25 and is used to compute the execution error matrix  $Q$ . Notice that for the first correction the most likely direction is in the  $XZ$  plane, but the maximum diagonal of the execution error covariance matrix is the term  $Q_{22}$  associated with uncertainties in  $\dot{Y}$ . This is explained by the execution error model and the error variances used for the particular run. For the specified quantities, the pointing errors are actually more severe than the total of the proportionality and resolution errors. For small angular pointing errors, the execution error

TABLE 25.- GUIDANCE CORRECTION INFORMATION FOR BASE ERROR  
ANALYSIS RUN

Correction no. Data	At 5 days 1	At 30 days 2	At 197 days 3
$E[ \Delta V ]$ , m/sec	8.638	0.079	0.744
$\sigma_{ \Delta }$ , m/sec	6.287	0.037	0.562
"E[ $\Delta V$ ]", m/sec	$\dot{X} = 7.716$ $\dot{Y} = -0.342$ $\dot{Z} = -3.866$	$\dot{X} = 0.066$ $\dot{Y} = 0.026$ $\dot{Z} = -0.036$	$\dot{X} = 0.572$ $\dot{Y} = -0.298$ $\dot{Z} = 0.370$
$\gamma_{11}$ , m <sup>2</sup> /sec <sup>2</sup>	$1.311 \times 10^{-3}$	$2.745 \times 10^{-4}$	$2.492 \times 10^{-4}$
$\gamma_{22}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.725 \times 10^{-3}$	$4.413 \times 10^{-5}$	$8.776 \times 10^{-5}$
$\gamma_{33}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.123 \times 10^{-3}$	$8.201 \times 10^{-5}$	$1.205 \times 10^{-4}$
$\eta_{14}$	$6.184 \times 10^6$	$3.358 \times 10^6$	$5.286 \times 10^5$
$\eta_{15}$	$-8.658 \times 10^6$	$-9.124 \times 10^6$	$-6.544 \times 10^5$
$\eta_{16}$	$6.051 \times 10^3$	$-1.248 \times 10^5$	$1.299 \times 10^2$
$\eta_{24}$	$-2.478 \times 10^7$	$-1.630 \times 10^7$	$-3.142 \times 10^5$
$\eta_{25}$	$-6.089 \times 10^6$	$-9.002 \times 10^6$	$-2.543 \times 10^5$
$\eta_{26}$	$-4.695 \times 10^6$	$-6.329 \times 10^6$	$-7.351 \times 10^5$

due to pointing is essentially orthogonal to the direction of the correction. Thus, the assumption of the first effective midcourse given by " $E [\Delta V]$ " and the specified variances given for the execution errors leads to a maximum execution uncertainty in a direction orthogonal to the maneuver. The  $3\sigma$  values for the execution error at the first correction are given by  $3\sigma_{\dot{X}} = 108$ ,  $3\sigma_{\dot{Y}} = 186$ , and  $3\sigma_{\dot{Z}} = 168$  mm/sec. These quantities are subsequently added to the navigation uncertainty covariance matrix for further propagation.

The bottom of table 25 gives the more important terms of the variation matrix  $\eta$ , which relates uncertainties in the impact parameter plane targeting conditions to uncertainties in the state vector at the time of the midcourse correction. At five days, the largest term is  $\eta_{24} = -2.478 \times 10^7$ , which means that a 1 m/sec error in  $\dot{X}$  at the time of the first correction maps into an error of 24,780 km in B·R when the Martian sphere of influence is encountered. Similarly, since  $\eta_{15} = -8.658 \times 10^6$ , a 1 m/sec error in  $\dot{Y}$  at the first correction time maps into an error of 8658 km in B·T at the target. The size of these sensitivities along the trajectory should be important for determining an optimum midcourse correction schedule. They are used, as would be expected, in determining target condition uncertainties after midcourse corrections.

The expected value of the second correction magnitude is 79 mm/sec. The standard deviation  $\sigma$  of this magnitude is 37 mm/sec. It should be stressed that these statistical computations for the second (and subsequent) midcourse corrections are conditioned on the preceding maneuvers. The execution error matrix for the first maneuver, which is the prime contributor in determining the most likely magnitude of the second correction, is based on the first effective midcourse maneuver " $E [\Delta V_1]$ ". Thus, the magnitudes for the second and third corrections should be interpreted as conditional distributions, given that preceding corrections are described by the effective correction vectors " $E [\Delta V]$ ". A determination of the statistical properties of subsequent midcourse maneuvers independent of any assumptions about corrections would require the simulation mode and a Monte Carlo set of runs.

Both the second and third maneuvers in the example run were relatively small. For small  $\Delta V$ 's, the most significant execution error, at least for the execution error statistics defined in the example run, is the resolution error. Since this error is along the direction of the commanded  $\Delta V$ , the  $\tilde{Q}$  terms (which are the diagonals of the execution error matrix) for the last two corrections show that the maximum execution uncertainty is along the vector of the effective midcourse correction " $E(\Delta V)$ ". Since the proportionality and pointing errors, both of which are a function of the magnitude of the maneuver, are very small for each of the last two corrections, there is not a significant difference between the sizes of the last two execution errors.

Table 26 gives the target condition uncertainties for the base error analysis run. Recall that these numbers answer the mission analysis question, "how close will the trajectory satisfy the nominal target conditions?" The first column displays the propagation of the injection covariance into the impact parameter plane. The initial target condition uncertainties, based on the injection errors, are  $3\sigma_{B.T} = 163\ 463\text{ km}$  and  $3\sigma_{B.R} = 778\ 078\text{ km}$ .

The next two rows define the major and minor axes of the  $3\sigma$  ellipse in the B-plane. The last row defines the orientation of the major axis of the  $3\sigma$  ellipse in the B-plane, where the angle is measured counterclockwise from the T-axis.

The numbers given in table 26 are self-explanatory and indicate how each successive midcourse maneuver reduces the likely actual trajectory deviations from the specified nominal target conditions. Notice that after the second correction, the uncertainty in B.R is almost forty times as great as the likely uncertainty in B.T. A closer inspection of the process explains why. From the preceding table it can be seen that the  $3\sigma$  values for the execution error at 30 days are given by 49.5, 19.8, and 27.3 mm/sec for the  $\dot{X}$ ,  $\dot{Y}$ , and  $\dot{Z}$  components respectively. These numbers result from using the effective midcourse correction " $E[\Delta V_2]$ " to compute the matrix  $\tilde{Q}$ . Assume first that the only error at the second correction was a  $+3\sigma$  or 49.5 mm/sec error in  $\dot{X}$ . Using the variation matrix components  $\eta_{14}$ , and  $\eta_{24}$ , the resulting impact parameter plane error would be approximately 170 km in B.T and 800 km in B.R. A full detailed analysis, including the correlation errors in the execution process, would demonstrate that for the given example problem, corrections at 5 days and 30 days would result in B.R errors roughly forty times as large as the errors in B.T.

TABLE 26.- TARGET CONDITION UNCERTAINTIES FOR BASE  
ERROR ANALYSIS RUN

Correction Data	After injection	After 1st midcourse	After 2nd midcourse	After 3rd midcourse
$3\sigma_{B \cdot T}$ , km	163 463	1692	22.2	43.7
$3\sigma_{B \cdot R}$ , km	778 078	3451	818	33.4
$3\sigma$ maximum eigenvalue, km	793 710	3452	818	52.6
$3\sigma$ minimum eigenvalue, km	46 361	1691	19.1	16.2
Orientation <sup>a</sup> of ellipse in B-plane, deg	+78.6	+88.7	+90.8	+35.7
<sup>a</sup> Angle measured is to major axis from T-axis (positive is counterclockwise).				

From table 26 it is apparent that the third midcourse correction, which occurs 197 days after injection, significantly shifts the orientation of the  $3\sigma$  uncertainty ellipse in the impact parameter plane. Also, even though the B.T uncertainty is actually increased by the third correction, this final midcourse reduces the B.R errors to an acceptable level.

2. Other Mars runs.- A number of additional error analysis runs based on the same nominal trajectory to Mars were made to demonstrate changes in key values. The first additional run varied only the execution error model and the number of guidance corrections. The third correction was omitted altogether to show what happens to the planetary approach navigation uncertainties when they are uninfluenced by a recent midcourse correction.

The change in the execution error model did not result in any significant navigation or target condition uncertainty shifts. However, as was expected, the navigation uncertainties in the neighborhood of Mars did exhibit measureable change. Table 27 shows the late navigation uncertainties for the error analysis mode run without the third midcourse correction and figure 13 compares a specific position component uncertainty during planetary approach, with and without a third midcourse correction 197 days after injection. The slight early difference between the two is due to using the auxiliary execution error model discussed in a previous section.

Table 27 shows even more dramatically the typical position and velocity uncertainty behavior that occurs as the vehicle passes through closest approach to the target planet. Before entering the sphere of influence of Mars, the orbit determination uncertainties are more or less constant as a result of navigation that has taken place over the long period of time since the second midcourse correction. About 500 000 km from Mars and approximately 204.8 days after injection, the velocity uncertainties begin to change. At a distance of 291 000 km from Mars, the  $3\sigma$  velocity uncertainties have increased significantly to 2.58, 4.05, and 1.17 mm/sec. Another measurement is made when the spacecraft is 33 000 km from Mars, and after this range rate measurement has been filtered, the  $3\sigma$  velocity uncertainties are still 60.3, 46.5, and 40.5 mm/sec. The vehicle is rapidly accelerating and, with no more measurements being processed, these velocity uncertainties increase to about 3 m/sec at periaresis. A measurement taken when the hypothetical vehicle is 236 000 km from Mars on an outgoing hyperbola only reduces the  $3\sigma$  velocity uncertainties to 2.13, 1.59, and 1.98 m/sec.

TABLE 27.- LATE NAVIGATION UNCERTAINTIES FOR BASE RUN WITHOUT 3RD MIDCOURSE CORRECTION

TRAJECTORY TIME, DAYS	DATA	MEAS NO.	KIND	POSITION UNCERTAINTIES, km <sup>2</sup>			VELOCITY UNCERTAINTIES, m <sup>2</sup> /sec <sup>2</sup>		
				$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
194.5 <sup>+</sup>		56	D <sub>2</sub>	7.358 x 10 <sup>1</sup>	2.552 x 10 <sup>1</sup>	1.113 x 10 <sup>1</sup>	2.201 x 10 <sup>-7</sup>	7.429 x 10 <sup>-7</sup>	1.047 x 10 <sup>-7</sup>
199.3 <sup>+</sup>		58	D <sub>1</sub>	7.555 x 10 <sup>1</sup>	2.391 x 10 <sup>1</sup>	1.037 x 10 <sup>1</sup>	1.743 x 10 <sup>-7</sup>	7.211 x 10 <sup>-7</sup>	9.919 x 10 <sup>-8</sup>
202.3 <sup>+</sup>		59	D <sub>1</sub>	7.688 x 10 <sup>1</sup>	2.381 x 10 <sup>1</sup>	1.009 x 10 <sup>1</sup>	1.519 x 10 <sup>-7</sup>	7.231 x 10 <sup>-7</sup>	9.724 x 10 <sup>-8</sup>
b 203.5 <sup>+</sup>		60	D <sub>1</sub>	7.726 x 10 <sup>1</sup>	2.277 x 10 <sup>1</sup>	9.894	1.401 x 10 <sup>-7</sup>	7.045 x 10 <sup>-7</sup>	9.485 x 10 <sup>-8</sup>
c 204.5 <sup>+</sup>		61	D <sub>1</sub>	7.758 x 10 <sup>1</sup>	2.180 x 10 <sup>1</sup>	9.739	1.305 x 10 <sup>-7</sup>	6.871 x 10 <sup>-7</sup>	9.273 x 10 <sup>-8</sup>
204.8 <sup>+</sup>		62	D <sub>3</sub>	7.727 x 10 <sup>1</sup>	2.033 x 10 <sup>1</sup>	9.550	1.886 x 10 <sup>-7</sup>	7.970 x 10 <sup>-7</sup>	5.826 x 10 <sup>-8</sup>
d 205.5 <sup>+</sup>		63	D <sub>1</sub>	7.139 x 10 <sup>1</sup>	1.754 x 10 <sup>1</sup>	8.686	7.313 x 10 <sup>-7</sup>	1.841 x 10 <sup>-6</sup>	1.550 x 10 <sup>-7</sup>
e 206.5 <sup>+</sup>		64	D <sub>1</sub>	8.359	9.349	3.237	4.054 x 10 <sup>-4</sup>	2.394 x 10 <sup>-4</sup>	1.829 x 10 <sup>-4</sup>
206.68		C <sub>A</sub>	----	2.089	1.317 x 10 <sup>1</sup>	2.654	6.881 x 10 <sup>-1</sup>	1.022	2.181
f 207.5 <sup>-</sup>		64	D <sub>1</sub>	3.400 x 10 <sup>3</sup>	2.073 x 10 <sup>3</sup>	7.791 x 10 <sup>3</sup>	5.451 x 10 <sup>-1</sup>	3.671 x 10 <sup>-1</sup>	1.129
207.5 <sup>+</sup>		65		3.095 x 10 <sup>3</sup>	1.657 x 10 <sup>3</sup>	2.801 x 10 <sup>3</sup>	5.023 x 10 <sup>-1</sup>	2.733 x 10 <sup>-1</sup>	4.369 x 10 <sup>-1</sup>
<sup>a</sup> Different execution error model.									
<sup>b</sup> Distance from Mars at 203.5 = 793 000 km.									
<sup>c</sup> Distance from Mars at 204.5 = 543 000 km.									
<sup>d</sup> Distance from Mars at 205.5 = 291 000 km.									
<sup>e</sup> Distance from Mars at 206.5 = 33 000 km.									
<sup>f</sup> Distance from Mars at 207.5 = 236 000 km.									

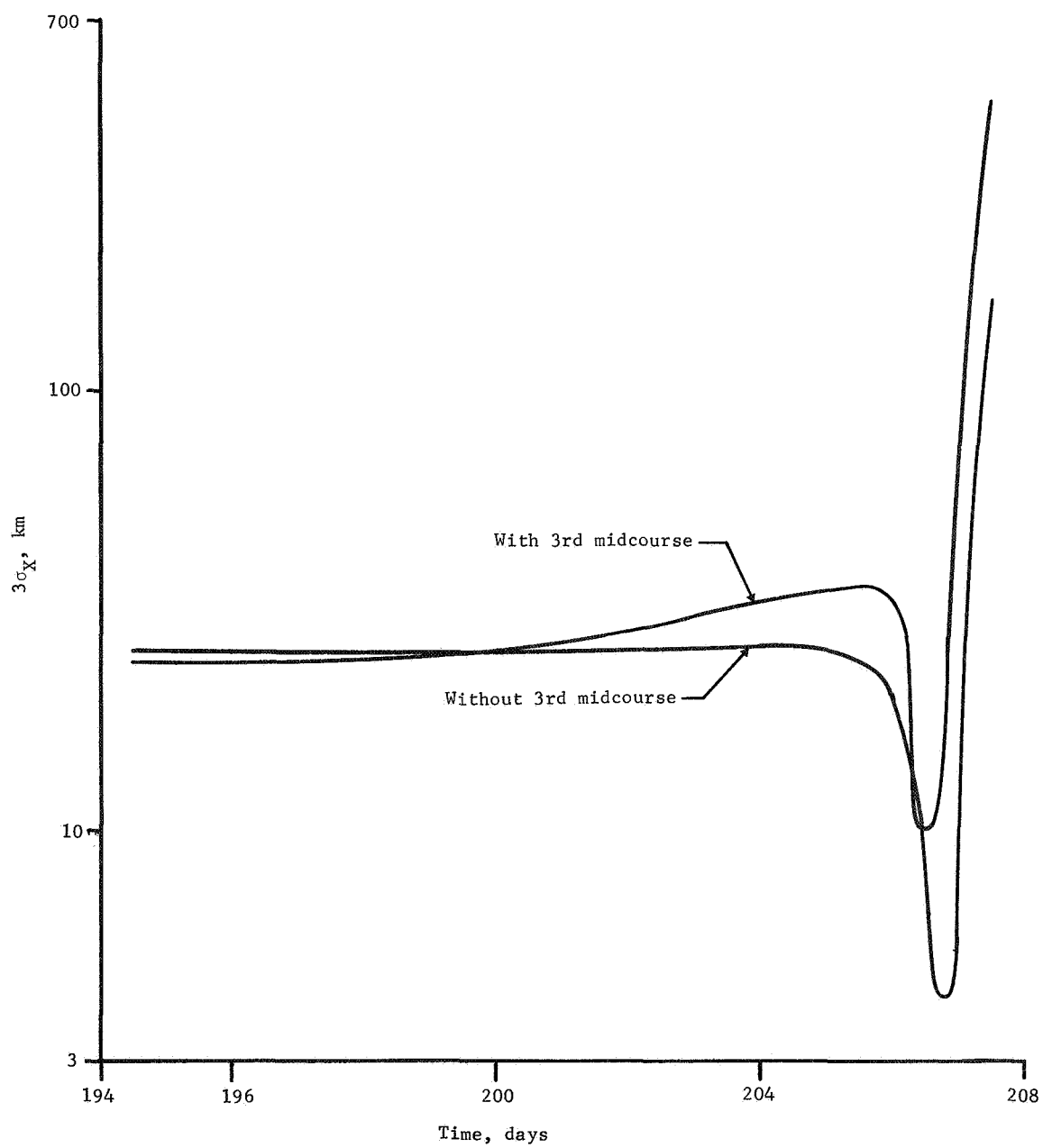


Figure 13.- Comparison of Late Navigation Uncertainties with and without 3<sup>rd</sup> Midcourse



The position uncertainties, on the other hand, drop dramatically as Mars is approached. At periapsis the  $3\sigma$  position uncertainties are given by 4.35, 10.8, and 4.9 km, considerably less than they were during the heliocentric phase of the trajectory. Just after periapsis, however, the velocity uncertainties that were building up on the incoming hyperbola cause the position uncertainties to skyrocket. After processing the range rate measurement made when the vehicle is almost a day past periapsis, the  $3\sigma$  position uncertainties are 165, 123, and 160 km.

The next error analysis run involved increasing the assumed measurement accuracies for both range and range rate by a factor of three. The  $3\sigma$  measurement inaccuracies were 3 mm/sec for range rate and 3 m for range. Tables 28 and 29 present the navigation uncertainties for the higher accuracy run. Figure 14 is a graph comparing the early navigation uncertainties for the base run and the higher accuracy run. It shows, in general, how higher accuracy measurements improve the convergence of the orbit determination process. Some interesting features of this comparison are worth noting. After one measurement, for example, the navigation uncertainties are virtually equal. The higher accuracy measurements only pay off when several measurements have been processed. By the time of the first guidance correction, the  $3\sigma$  navigation uncertainties for the high accuracy run are roughly one-third those for the base run. Studies similar to this one indicate that if a system is completely observable, the increased navigation accuracy due to increased measurement accuracy is roughly one to one. With a partially observable system the above statement is not true.

Another error analysis mode run was made with an injection covariance matrix that was still diagonal, but whose terms were four times as large as those in the base run. The  $3\sigma$  injection errors were thus twice as large. As was expected, the expected magnitude of the first midcourse almost exactly doubled; it was 17.275 m/sec. The remaining midcourse magnitudes were also approximately twice as large. The run was made with the higher accuracy measurements assumed. Table 30, which presents scattered navigation uncertainties for the poor injection run, should be compared with tables 28 and 29 to obtain the influence of a poorer injection covariance on the processed orbit determination uncertainties. Position and velocity uncertainties, although initially larger, are filtered to approximately the same level despite the poorer injection.

TABLE 28.- EARLY NAVIGATION UNCERTAINTIES FOR HIGHER ACCURACY BASE RUN

Data		Meas No.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
Trajectory time, days	$\sigma_X^2$			$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	
0	0			1.000	1.000	1.000	9.000	9.000	9.000
0.2 <sup>+</sup>	1		D <sub>1</sub> , D <sub>1</sub>	2.130 x 10 <sup>2</sup>	1.626 x 10 <sup>3</sup>	1.855 x 10 <sup>2</sup>	7.573 x 10 <sup>-1</sup>	3.503	5.636 x 10 <sup>-1</sup>
0.9 <sup>+</sup>	3		D <sub>3</sub> , D <sub>3</sub>	2.057 x 10 <sup>-1</sup>	1.179 x 10 <sup>-1</sup>	3.830 x 10 <sup>-1</sup>	1.494 x 10 <sup>-4</sup>	9.816 x 10 <sup>-5</sup>	1.654 x 10 <sup>-4</sup>
1.6 <sup>+</sup>	5		D <sub>2</sub> , D <sub>2</sub>	5.556 x 10 <sup>-2</sup>	1.433 x 10 <sup>-2</sup>	9.967 x 10 <sup>-2</sup>	2.763 x 10 <sup>-5</sup>	1.069 x 10 <sup>-5</sup>	3.283 x 10 <sup>-5</sup>
2.2 <sup>+</sup>	7		D <sub>1</sub> , D <sub>1</sub>	5.199 x 10 <sup>-3</sup>	1.061 x 10 <sup>-2</sup>	5.836 x 10 <sup>-3</sup>	3.740 x 10 <sup>-7</sup>	3.916 x 10 <sup>-7</sup>	3.885 x 10 <sup>-7</sup>
3.6 <sup>+</sup>	11		D <sub>2</sub> , D <sub>2</sub>	7.640 x 10 <sup>-3</sup>	1.229 x 10 <sup>-2</sup>	9.229 x 10 <sup>-3</sup>	1.259 x 10 <sup>-7</sup>	1.299 x 10 <sup>-7</sup>	1.620 x 10 <sup>-7</sup>
4.6 <sup>+</sup>	14		D <sub>2</sub> , D <sub>2</sub>	8.958 x 10 <sup>-3</sup>	1.517 x 10 <sup>-2</sup>	1.034 x 10 <sup>-2</sup>	7.821 x 10 <sup>-8</sup>	9.536 x 10 <sup>-8</sup>	9.732 x 10 <sup>-8</sup>
5.0 <sup>-</sup>	G <sub>1</sub>		----	9.725 x 10 <sup>-3</sup>	1.686 x 10 <sup>-2</sup>	1.229 x 10 <sup>-2</sup>	7.297 x 10 <sup>-8</sup>	9.117 x 10 <sup>-8</sup>	9.560 x 10 <sup>-8</sup>
5.0 <sup>+</sup>							<sup>a</sup> 1.607 x 10 <sup>-3</sup>	4.838 x 10 <sup>-3</sup>	4.032 x 10 <sup>-3</sup>
5.3 <sup>+</sup>	16		D <sub>1</sub>	4.667 x 10 <sup>-1</sup>	2.041	1.409	6.829 x 10 <sup>-4</sup>	3.031 x 10 <sup>-3</sup>	2.091 x 10 <sup>-3</sup>
6.7 <sup>+</sup>	17		D <sub>2</sub> , D <sub>2</sub>	1.154 x 10 <sup>-1</sup>	2.630 x 10 <sup>-1</sup>	1.943 x 10 <sup>-1</sup>	5.356 x 10 <sup>-4</sup>	1.222 x 10 <sup>-3</sup>	9.008 x 10 <sup>-4</sup>
8.0 <sup>+</sup>	18		D <sub>3</sub> , D <sub>3</sub>	1.072 x 10 <sup>-1</sup>	3.171 x 10 <sup>-1</sup>	3.656	1.607 x 10 <sup>-4</sup>	4.736 x 10 <sup>-4</sup>	5.458 x 10 <sup>-5</sup>

<sup>a</sup>Different execution error model.

<sup>a</sup>Different execution error model.

TABLE 29.- LATE NAVIGATION UNCERTAINTIES FOR HIGH ACCURACY RUN WITHOUT 3RD MIDCOURSE CORRECTION

Data Trajectory time, days	Meas no.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
194.5 <sup>+</sup>	56	D <sub>2</sub>	1.604 x 10 <sup>-1</sup>	3.519	2.881	2.916 x 10 <sup>-8</sup>	9.203 x 10 <sup>-8</sup>	1.897 x 10 <sup>-8</sup>
199.3 <sup>+</sup>	58	D <sub>1</sub>	1.636 x 10 <sup>-1</sup>	3.022	2.865	2.342 x 10 <sup>-8</sup>	9.098 x 10 <sup>-8</sup>	1.798 x 10 <sup>-8</sup>
202.3 <sup>+</sup>	59	D <sub>1</sub>	1.654 x 10 <sup>-1</sup>	2.904	2.883	2.038 x 10 <sup>-8</sup>	9.199 x 10 <sup>-8</sup>	1.746 x 10 <sup>-8</sup>
203.5 <sup>+</sup>	60	D <sub>1</sub>	1.658 x 10 <sup>-1</sup>	2.718	2.880	1.902 x 10 <sup>-8</sup>	9.079 x 10 <sup>-8</sup>	1.719 x 10 <sup>-8</sup>
204.5 <sup>+</sup>	61	D <sub>1</sub>	1.661 x 10 <sup>-1</sup>	2.559	2.879	1.791 x 10 <sup>-8</sup>	8.971 x 10 <sup>-8</sup>	1.697 x 10 <sup>-8</sup>
204.8 <sup>+</sup>	62	D <sub>3</sub>	1.664 x 10 <sup>-1</sup>	2.360	2.853	2.674 x 10 <sup>-8</sup>	1.118 x 10 <sup>-7</sup>	1.313 x 10 <sup>-8</sup>
205.5 <sup>+</sup>	63	D <sub>1</sub>	1.539 x 10 <sup>-1</sup>	2.041	2.564	1.059 x 10 <sup>-7</sup>	3.141 x 10 <sup>-7</sup>	3.710 x 10 <sup>-8</sup>
206.5 <sup>+</sup>	64	D <sub>1</sub>	1.787	1.545	1.073	7.183 x 10 <sup>-5</sup>	4.291 x 10 <sup>-5</sup>	5.908 x 10 <sup>-5</sup>
206.68	C <sub>A</sub>	----	5.925 x 10 <sup>-1</sup>	2.341	7.577 x 10 <sup>-1</sup>	1.572 x 10 <sup>-1</sup>	2.803 x 10 <sup>-1</sup>	3.688 x 10 <sup>-1</sup>
207.5 <sup>-</sup>	64	D <sub>1</sub>	1.077 x 10 <sup>-3</sup>	4.186 x 10 <sup>-2</sup>	1.289 x 10 <sup>-3</sup>	1.715 x 10 <sup>-1</sup>	7.097 x 10 <sup>-2</sup>	1.872 x 10 <sup>-1</sup>
207.5 <sup>+</sup>	65		7.678 x 10 <sup>-2</sup>	4.108 x 10 <sup>-2</sup>	6.950 x 10 <sup>-2</sup>	1.246 x 10 <sup>-1</sup>	6.779 x 10 <sup>-2</sup>	1.084 x 10 <sup>-1</sup>

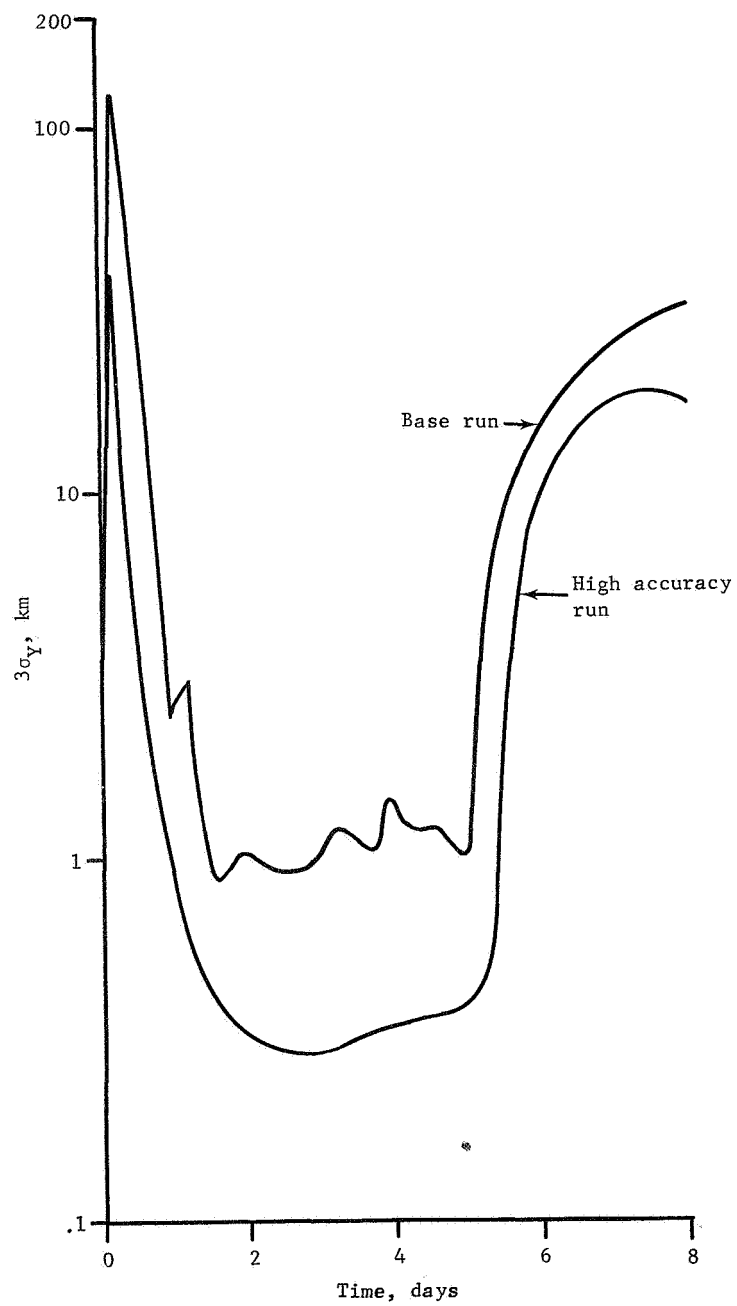


Figure 14.- Navigation Uncertainties for Error Analysis Run

TABLE 30.- SCATTERED NAVIGATION UNCERTAINTIES FOR POOR INJECTION RUN WITHOUT 3RD MIDCOURSE CORRECTION BUT WITH HIGH ACCURACY

Trajectory time, days	Data	Meas no.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
				$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
0		0	----	4.000	4.000	4.000	$3.600 \times 10^1$	$3.600 \times 10^1$	$3.600 \times 10^1$
0.2 <sup>+</sup>		1	D <sub>1</sub> , D <sub>1</sub>	$8.519 \times 10^2$	$6.503 \times 10^3$	$7.419 \times 10^2$	3.028	$1.401 \times 10^1$	2.255
3.6 <sup>+</sup>		11	D <sub>2</sub> , D <sub>2</sub>	$7.645 \times 10^{-3}$	$1.230 \times 10^{-2}$	$9.236 \times 10^{-3}$	$1.261 \times 10^{-7}$	$1.300 \times 10^{-7}$	$1.623 \times 10^{-7}$
5.0 <sup>-</sup>		G <sub>1</sub>		$9.729 \times 10^{-3}$	$1.686 \times 10^{-2}$	$1.230 \times 10^{-2}$	$7.303 \times 10^{-8}$	$9.121 \times 10^{-8}$	$9.578 \times 10^{-8}$
5.0 <sup>+</sup>							$5.468 \times 10^{-3}$	$1.935 \times 10^{-2}$	$1.589 \times 10^{-2}$
8.0 <sup>+</sup>		18	D <sub>3</sub> , D <sub>3</sub>	$1.269 \times 10^1$	$3.747 \times 10^1$	4.125	$1.904 \times 10^{-4}$	$5.598 \times 10^{-4}$	$6.154 \times 10^{-5}$
194.5 <sup>+</sup>		56	D <sub>2</sub>	$1.847 \times 10^1$	3.750	3.596	$3.073 \times 10^{-8}$	$9.577 \times 10^{-8}$	$2.228 \times 10^{-8}$
203.5 <sup>+</sup>		60	D <sub>1</sub>	$1.881 \times 10^1$	2.778	3.641	$2.007 \times 10^{-8}$	$9.507 \times 10^{-8}$	$2.004 \times 10^{-8}$
204.8 <sup>+</sup>		62	D <sub>3</sub>	$1.882 \times 10^1$	2.393	3.613	$2.853 \times 10^{-8}$	$1.192 \times 10^{-7}$	$1.604 \times 10^{-8}$
206.5 <sup>+</sup>		64	D <sub>1</sub>	2.054	1.623	1.351	$7.775 \times 10^{-5}$	$4.669 \times 10^{-5}$	$7.404 \times 10^{-5}$
206.68		C <sub>A</sub>	----	$7.287 \times 10^{-1}$	2.523	$9.264 \times 10^{-1}$	$1.840 \times 10^{-1}$	$3.399 \times 10^{-1}$	$3.903 \times 10^{-1}$
207.5 <sup>-</sup>		64	D <sub>1</sub>	$1.349 \times 10^3$	$4.732 \times 10^2$	$1.355 \times 10^3$	$2.146 \times 10^{-1}$	$7.913 \times 10^{-2}$	$1.970 \times 10^{-1}$
207.5 <sup>+</sup>		65		$8.824 \times 10^2$	$4.722 \times 10^2$	$7.987 \times 10^2$	$1.432 \times 10^{-1}$	$7.791 \times 10^{-2}$	$1.244 \times 10^{-1}$

Two error analysis mode runs were also made under the assumption that some dynamic or process noise existed. For the first run, called the low dynamic noise run in table 31, the diagonal process noise matrix was computed over each time interval with a  $k$  value of  $10^{-22} \text{ km}^2/\text{sec}^4$  in all three acceleration components. This roughly corresponds to an unmodelled acceleration ( $3\sigma$ ) of  $3 \times 10^{-5} \text{ mm/sec}^2$ . The next run, called the high dynamic noise run, used a  $k$  value of  $10^{-18} \text{ km}^2/\text{sec}^4$ , which represents a  $3\sigma$  unmodelled acceleration of  $3 \times 10^{-3} \text{ mm/sec}^2$ .

The purpose of the two runs was to determine the increased magnitude of the velocity corrections when some random and bias accelerations were present along the trajectory. From table 31 it is obvious that the first level of dynamic noise does not significantly change the likely magnitudes of the midcourse maneuvers except for the third correction. The most likely value for the third correction magnitude is now 2.418 m/sec, which reflects the modeling of the unknown accelerations over the long heliocentric phase between the two corrections.

For the high dynamic noise run (table 32) the likely magnitude of the first correction remains more or less the same. However, the second correction requires significantly more fuel (3.521 m/sec for  $|\Delta V|$ ) and the third correction, of likely magnitude 223.17 m/sec, is horrendous. In some general sense, then, these two runs indicate that fuel requirements remain approximately the same if the unmodelled accelerations are of the order of  $10^{-5} \text{ mm/sec}^2$ . However, if there are unmodelled accelerations as high as  $10^{-3} \text{ mm/sec}^2$  acting along the trajectory -- particularly if these accelerations are constant biases -- then the fuel requirements for the midcourse maneuvers are much higher.

No target condition uncertainties are shown for the dynamic noise runs because their meaning is obscure. The artificial addition of diagonal dynamic noise, although it can result in a feel for likely correction magnitudes due to unmodelled accelerations, changes the geometry of the processed covariance matrices in an arbitrary fashion. This, in turn, results in impact parameter plane dispersions that reflect the arbitrary addition of the dynamic noise.

TABLE 31.- GUIDANCE CORRECTION INFORMATION FOR LOW DYNAMIC NOISE RUN

Correction No. Data	At 5 days 1	At 30 days 2	At 197 days 3
$E [ \Delta V ]$ , m/sec	8.638	0.087	2.418
$\sigma_{ \Delta V }$ , m/sec	6.287	0.039	1.056
"E $[\Delta V]$ ", m/sec	$\dot{X} = 7.716$ $\dot{Y} = 0.342$ $\dot{Z} = -3.866$	$\dot{X} = 0.072$ $\dot{Y} = 0.029$ $\dot{Z} = -0.392$	$\dot{X} = 1.809$ $\dot{Y} = -1.221$ $\dot{Z} = 1.039$
$\tilde{Q}_{11}$ , m <sup>2</sup> /sec <sup>2</sup>	$1.311 \times 10^{-3}$	$2.761 \times 10^{-4}$	$3.657 \times 10^{-4}$
$\tilde{Q}_{22}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.728 \times 10^{-3}$	$4.337 \times 10^{-5}$	$3.258 \times 10^{-4}$
$\tilde{Q}_{33}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.123 \times 10^{-3}$	$8.131 \times 10^{-5}$	$3.166 \times 10^{-4}$
$\eta_{15}$	$-8.658 \times 10^6$	$-9.124 \times 10^6$	$-6.544 \times 10^5$
$\eta_{24}$	$-2.478 \times 10^7$	$-1.630 \times 10^7$	$-3.142 \times 10^5$

TABLE 32.- GUIDANCE CORRECTION INFORMATION FOR HIGH DYNAMIC NOISE RUN

Correction No. Data	At 5 days 1	At 30 days 2	At 197 days 3
$E  \Delta V $ , m/sec	3.671	3.521	223.172
$\sigma_{ \Delta V }$ , m/sec	6.285	1.408	92.664
"E $[V]$ ", m/sec	$\dot{X} = 7.747$ $\dot{Y} = -0.343$ $\dot{Z} = -3.882$	$\dot{X} = 3.519$ $\dot{Y} = 0.047$ $\dot{Z} = -0.125$	$\dot{X} = 152.997$ $\dot{Y} = -75.765$ $\dot{Z} = 143.727$
$\tilde{Q}_{11}$ , m <sup>2</sup> /sec <sup>2</sup>	$1.318 \times 10^{-3}$	$4.498 \times 10^{-4}$	1.414
$\tilde{Q}_{22}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.755 \times 10^{-3}$	$6.199 \times 10^{-4}$	2.226
$\tilde{Q}_{33}$ , m <sup>2</sup> /sec <sup>2</sup>	$3.147 \times 10^{-3}$	$6.198 \times 10^{-4}$	1.540
$\eta_{15}$	$-8.658 \times 10^6$	$-9.124 \times 10^6$	$-6.544 \times 10^5$
$\eta_{24}$	$-2.478 \times 10^7$	$-1.630 \times 10^7$	$-3.142 \times 10^5$

The next example run, called the auxiliary error analysis run on the tables, again used 65 measurements. Now, however, 61 of these measurements were from the same Earth-based station (Goldstone) and four of the observations, taken near the planet Mars, were assumed to come from onboard sensors. Table 33 gives the detailed inputs for the auxiliary error analysis run. The nominal interplanetary trajectory was still the same as given in table 22.

Table 34 tabulates the early navigation uncertainties along the auxiliary run and can be compared with table 23 to establish whether or not three different tracking stations operating one-third as often are better than one station in reducing navigation uncertainties. Table 35 features the processing of hypothetical onboard measurements when the vehicle was near Mars.

For both the base run and the auxiliary example run of the error analysis mode, 15 separate measurements of range and range rate were made before the first midcourse correction. Taking five measurements each from Goldstone, Madrid, and Canberra, the  $3\sigma$  navigation errors five days after injection were 0.87, 1.17, and 0.98 km for the position uncertainties, and 2.43, 2.71, and 2.76 mm/sec for the velocity uncertainties. When all 15 observations were assumed to have been made by Goldstone, the comparable position uncertainties were 1.91, 2.52, and 2.40 km, while the  $3\sigma$  velocity errors were 4.95, 5.79, and 5.85 mm/sec. Thus, even though the same number of measurements of the same accuracy were taken for both runs, when the observations came from three different stations the  $3\sigma$  uncertainties at the first midcourse were less than half as large. This phenomenon was noticed repeatedly during the test runs and is doubtless related to the concept of observability as well as the geometry of the specific problem involved.

The final error analysis example run for an interplanetary flight from Earth to Mars used an augmented state vector. The program was exercised in the error analysis mode with  $IAUG = 11$ , meaning the state vector had 17 components including three Goldstone station location biases, biases in  $\mu$  for the Sun and Mars, range and range rate biases from Goldstone, and biases in all four onboard measurements. Except for the inclusion of the augmented state vector, all input conditions were the same as for the auxiliary error analysis run just discussed.



TABLE 33.- DIFFERENT MEASUREMENT SCHEDULE FOR AUXILIARY ERROR ANALYSIS RUN

Goldstone measured range and range rate 15 times at 0.2, 0.4, 0.6, 1.2, 1.4, 1.6, 2.2, 2.4, 2.6, 3.2, 3.4, 3.6, 4.2, 4.4, 4.6
Goldstone measured range rate 46 times at 5.3, 6.3, 7.3, 8.3, 9.3, 10.3, 12.3, 14.3, 16.3, 18.3, 20.3, 22.3, 24.3, 26.3, 28.3, 30.3, 32.3, 34.3, 36.3, 38.3, 40.3, 43.5, 53.5, 63.5, 73.5, 83.5, 93.5, 103.5, 113.5, 123.5, 133.5, 143.5, 153.5, 163.5, 173.5, 183.5, 193.5, 195.5, 197.5, 199.5, 201.5, 203.5, 204.6, 205.6, 206.6, 207.6
Three star-planet angles measured 2 times at 205.4, 206.4
Apparent planet diameter measured 2 times at 205.8, 206.8
Total of 65 measurements
Reference stars were Canopus, Betelgeuse, Rigel
Star-planet measurement accuracies and apparent planet diameter accuracies were
Two guidance corrections, three variable B-plane policy, at 5 days and 30 days
$\sigma_{\text{angles}}^2 = 10^{-8} \text{ rad}^2$

TABLE 34.- EARLY NAVIGATION UNCERTAINTIES FOR AUXILIARY ERROR ANALYSIS RUN

Trajectory time, days	Data	Meas no.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
				$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
0		0		1.000		1.000	9.000	9.000	9.000
0.2 <sup>+</sup>		1	D <sub>1</sub> , D <sub>1</sub>	2.131 x 10 <sup>2</sup>	1.626 x 10 <sup>3</sup>	1.856 x 10 <sup>2</sup>	7.575 x 10 <sup>-1</sup>	3.504	5.639 x 10 <sup>-1</sup>
0.6 <sup>+</sup>		3	D <sub>1</sub> , D <sub>1</sub>	1.056 x 10 <sup>1</sup>	6.020	4.128 x 10 <sup>1</sup>	2.877 x 10 <sup>-3</sup>	1.268 x 10 <sup>-3</sup>	8.029 x 10 <sup>-3</sup>
2.6 <sup>+</sup>		9	D <sub>1</sub> , D <sub>1</sub>	2.407 x 10 <sup>-1</sup>	3.936 x 10 <sup>-1</sup>	3.260 x 10 <sup>-1</sup>	7.476 x 10 <sup>-6</sup>	8.100 x 10 <sup>-6</sup>	8.522 x 10 <sup>-6</sup>
4.6 <sup>+</sup>		15	D <sub>1</sub> , D <sub>1</sub>	3.356 x 10 <sup>-1</sup>	6.038 x 10 <sup>-1</sup>	5.371 x 10 <sup>-1</sup>	2.671 x 10 <sup>-6</sup>	3.721 x 10 <sup>-6</sup>	3.837 x 10 <sup>-6</sup>
5.0 <sup>-</sup>		G <sub>1</sub>	----	4.033 x 10 <sup>-1</sup>	7.116 x 10 <sup>-1</sup>	6.394 x 10 <sup>-1</sup>	2.675 x 10 <sup>-6</sup>	3.728 x 10 <sup>-6</sup>	3.833 x 10 <sup>-6</sup>
5.0 <sup>+</sup>							1.609 x 10 <sup>-3</sup>	4.841 x 10 <sup>-3</sup>	4.035 x 10 <sup>-3</sup>
5.3 <sup>+</sup>		16	D <sub>1</sub>	9.128 x 10 <sup>-1</sup>	2.826 x 10 <sup>-1</sup>	2.120	6.871 x 10 <sup>-4</sup>	3.038 x 10 <sup>-3</sup>	2.099 x 10 <sup>-3</sup>
6.3 <sup>+</sup>		17	D <sub>1</sub>	9.201	3.936 x 10 <sup>-1</sup>	2.739 x 10 <sup>-1</sup>	6.814 x 10 <sup>-4</sup>	3.039 x 10 <sup>-3</sup>	2.095 x 10 <sup>-3</sup>
10.3 <sup>+</sup>		21	D <sub>1</sub>	1.358 x 10 <sup>2</sup>	6.284 x 10 <sup>-2</sup>	4.395 x 10 <sup>2</sup>	6.515 x 10 <sup>-4</sup>	3.007 x 10 <sup>-3</sup>	2.076 x 10 <sup>-3</sup>
20.3 <sup>+</sup>		26	D <sub>1</sub>	6.609 x 10 <sup>2</sup>	3.631 x 10 <sup>3</sup>	3.380 x 10 <sup>3</sup>	4.315 x 10 <sup>-4</sup>	2.144 x 10 <sup>-3</sup>	1.850 x 10 <sup>-3</sup>
30.0 <sup>-</sup>		G <sub>2</sub>	----	5.191 x 10 <sup>2</sup>	3.559 x 10 <sup>3</sup>	7.299 x 10 <sup>3</sup>	1.370 x 10 <sup>-4</sup>	7.659 x 10 <sup>-4</sup>	1.392 x 10 <sup>-3</sup>
30.0 <sup>+</sup>							3.992 x 10 <sup>-4</sup>	8.140 x 10 <sup>-4</sup>	1.482 x 10 <sup>-3</sup>
40.3 <sup>+</sup>		36	D <sub>1</sub>	5.968 x 10 <sup>2</sup>	4.860 x 10 <sup>3</sup>	1.311 x 10 <sup>4</sup>	5.069 x 10 <sup>-5</sup>	5.072 x 10 <sup>-4</sup>	1.144 x 10 <sup>-3</sup>
133.5 <sup>+</sup>		46	D <sub>1</sub>	1.461 x 10 <sup>2</sup>	3.003 x 10 <sup>2</sup>	3.847 x 10 <sup>2</sup>	1.333 x 10 <sup>-6</sup>	8.524 x 10 <sup>-7</sup>	4.847 x 10 <sup>-7</sup>

TABLE 35.- LATE NAVIGATION UNCERTAINTIES FOR AUXILIARY ERROR ANALYSIS RUN

Data Trajectory time, days	Meas No.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
193.5 <sup>+</sup>	52	$\dot{D}_1$	1.918 x 10 <sup>2</sup>	5.908 x 10 <sup>1</sup>	1.486 x 10 <sup>2</sup>	3.789 x 10 <sup>-7</sup>	1.319 x 10 <sup>-6</sup>	2.345 x 10 <sup>-6</sup>
199.5 <sup>+</sup>	55	$\dot{D}_1$	1.914 x 10 <sup>2</sup>	3.663 x 10 <sup>1</sup>	1.304 x 10 <sup>2</sup>	2.593 x 10 <sup>-7</sup>	1.247 x 10 <sup>-6</sup>	2.517 x 10 <sup>-6</sup>
a203.5 <sup>+</sup>	57	$\dot{D}_1$	1.922 x 10 <sup>2</sup>	3.073 x 10 <sup>1</sup>	1.193 x 10 <sup>2</sup>	2.075 x 10 <sup>-7</sup>	1.245 x 10 <sup>-6</sup>	2.628 x 10 <sup>-6</sup>
b204.6 <sup>+</sup>	58	$\dot{D}_1$	1.922 x 10 <sup>2</sup>	2.841 x 10 <sup>1</sup>	1.163 x 10 <sup>2</sup>	1.920 x 10 <sup>-7</sup>	1.237 x 10 <sup>-6</sup>	2.657 x 10 <sup>-6</sup>
c205.4 <sup>-</sup>	59	Angles	1.929 x 10 <sup>2</sup>	2.899 x 10 <sup>1</sup>	1.140 x 10 <sup>2</sup>	7.899 x 10 <sup>-7</sup>	2.974 x 10 <sup>-6</sup>	3.820 x 10 <sup>-6</sup>
205.4 <sup>+</sup>			1.581 x 10 <sup>2</sup>	2.870 x 10 <sup>1</sup>	1.017 x 10 <sup>2</sup>	7.861 x 10 <sup>-7</sup>	2.670 x 10 <sup>-6</sup>	3.313 x 10 <sup>-6</sup>
d205.8 <sup>-</sup>	61	$\beta$	1.469 x 10 <sup>2</sup>	2.369 x 10 <sup>1</sup>	9.384 x 10 <sup>1</sup>	2.760 x 10 <sup>-6</sup>	7.024 x 10 <sup>-6</sup>	7.083 x 10 <sup>-6</sup>
205.8 <sup>+</sup>			1.469 x 10 <sup>2</sup>	2.369 x 10 <sup>1</sup>	9.384 x 10 <sup>1</sup>	2.760 x 10 <sup>-6</sup>	7.022 x 10 <sup>-6</sup>	7.083 x 10 <sup>-6</sup>
e206.4 <sup>-</sup>	62	Angles	1.500 x 10 <sup>2</sup>	2.483 x 10 <sup>1</sup>	8.955 x 10 <sup>1</sup>	4.406 x 10 <sup>-4</sup>	9.593 x 10 <sup>-4</sup>	6.342 x 10 <sup>-4</sup>
206.4 <sup>+</sup>			4.640 x 10 <sup>1</sup>	2.324 x 10 <sup>1</sup>	3.539 x 10 <sup>1</sup>	3.863 x 10 <sup>-4</sup>	5.040 x 10 <sup>-4</sup>	1.963 x 10 <sup>-4</sup>
f206.6 <sup>+</sup>	63	$\dot{D}_1$	1.832 x 10 <sup>1</sup>	4.177	1.942 x 10 <sup>1</sup>	3.871 x 10 <sup>-1</sup>	1.939 x 10 <sup>-1</sup>	1.042
g206.8 <sup>-</sup>	64	$\beta$	5.294 x 10 <sup>2</sup>	8.037 x 10 <sup>1</sup>	6.098 x 10 <sup>2</sup>	1.619	3.150 x 10 <sup>-1</sup>	1.427
206.8 <sup>+</sup>			5.291 x 10 <sup>2</sup>	8.037 x 10 <sup>1</sup>	6.092 x 10 <sup>2</sup>	1.618	3.149 x 10 <sup>-1</sup>	1.426
207.6 <sup>-</sup>	65	$\dot{D}_1$	1.141 x 10 <sup>4</sup>	2.127 x 10 <sup>3</sup>	1.054 x 10 <sup>4</sup>	1.421	2.820 x 10 <sup>-1</sup>	1.229
207.6 <sup>+</sup>			3.812 x 10 <sup>3</sup>	2.070 x 10 <sup>3</sup>	3.233 x 10 <sup>3</sup>	5.020 x 10 <sup>-1</sup>	2.703 x 10 <sup>-1</sup>	4.129 x 10 <sup>-1</sup>
<sup>a</sup> Distance from Mars at 203.5 = 793 000 km. <sup>b</sup> Distance from Mars at 204.6 = 518 000 km. <sup>c</sup> Distance from Mars at 205.4 = 317 000 km. <sup>d</sup> Distance from Mars at 205.8 = 215 000 km. <sup>e</sup> Distance from Mars at 206.4 = 61 000 km. <sup>f</sup> Distance from Mars at 206.6 = 5900 km. <sup>g</sup> Distance from Mars at 206.8 = 56 000 km.								

Table 36 contrasts the target condition uncertainties after injection and the two corrections for the augmented and non-augmented runs. The uncertainties in the various augmented parameters were quite small initially and their contribution to increasing the overall navigation uncertainty, as reflected in target condition dispersions, were slight as shown in the table. The discrepancy between the numbers presented in tables 36 and 23 is partially due to the higher navigation uncertainties, but is mostly the result of an extremely simple execution error model used early in the development of STEAP and subsequently discarded. The main point of table 36 is to confirm that reasonable uncertainty values for the augmented parameters result in only small changes in the target condition uncertainties. The specific hypothetical uncertainties for the 11 augmented biases, all of which were assumed to have mean zero, are given in the first line of table 39.

Tables 34, 35, 37, and 38 contrast the position and velocity navigation uncertainties resulting from the augmented and non-augmented runs. Figure 15 presents the contrast between them for a particular component of the navigation uncertainty. Examining tables 34 and 37, the effect of the augmented state is clear. After the processing of only a few measurements, when comparatively large navigation uncertainties are being reduced by the estimation algorithm, there is not a significant difference between the two. However, the non augmented state error analysis run assumes that the values for all the biases are known to be exactly zero, with no error, and thus reduces the position uncertainties to  $3\sigma$  values of 1.91, 2.52, and 2.40 km just before the first midcourse correction. The augmented state vector run, even though it is based on a "solve for" navigation algorithm, acknowledges the existence of uncertainties in the augmented parameters. Consequently, the reduction in the navigation uncertainties is inhibited by these uncertainties and the  $3\sigma$  position errors after processing 15 range and range rate measurements are higher. At the time of the first midcourse correction these higher navigation position uncertainties are given by 2.71, 2.81 and 3.75 km.

Comparing tables 35 and 38, one can see the total effect, over the entire example trajectory, of acknowledging the specified uncertainties in the augmented parameters. After processing the last range rate observation taken after the vehicle has passed Mars periapsis, the  $3\sigma$  position errors for the nonaugmented run are found to be 186, 135, and 168 km. For the augmented run made under exactly the same conditions, these final  $3\sigma$  position uncertainties are 256, 190, and 236 km, roughly 30 to 40 percent higher.

TABLE 36.- TARGET CONDITION UNCERTAINTIES FOR AUXILIARY ERROR  
ANALYSIS RUN, AUGMENTED VS NONAUGMENTED

Data	Nonaugmented <sup>a</sup>			Augmented <sup>a</sup>		
	After injection	After 1st midcourse	After 2nd midcourse	After injection	After 1st midcourse	After 2nd midcourse
$3\sigma_{B \cdot T}$ , km	163 463	1920	751	163 463	1920	757
$3\sigma_{B \cdot R}$ , km	778 078	3880	928	778 078	3881	930
3 $\sigma$ maximum eigenvalue, km	793 710	3880	943	793 710	3881	947
3 $\sigma$ minimum eigenvalue, km	46 361	1919	731	46 361	1920	736
Ellipse orientation in B-plane, deg	+78.6	+87.5	+73.3	+78.6	+89.2	+72.7
<sup>a</sup> This run uses different execution error model from base run.						

TABLE 37.- EARLY NAVIGATION UNCERTAINTIES FOR AUGMENTED RUN

Data Trajectory time, days	Meas No.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
0	0		1.000					
0.2 <sup>+</sup>	1	D <sub>1</sub> , $\dot{D}_1$	2.131 × 10 <sup>2</sup>	1.000	1.000	9.000	9.000	9.000
0.6 <sup>+</sup>	3	D <sub>1</sub> , $\dot{D}_1$	1.062 × 10 <sup>1</sup>	1.626 × 10 <sup>3</sup>	1.856 × 10 <sup>2</sup>	7.575 × 10 <sup>-1</sup>	3.504	5.639 × 10 <sup>-1</sup>
2.6 <sup>+</sup>	9	D <sub>1</sub> , $\dot{D}_1$	3.602 × 10 <sup>-1</sup>	6.031	4.140 × 10 <sup>1</sup>	2.895 × 10 <sup>-3</sup>	1.273 × 10 <sup>-3</sup>	8.066 × 10 <sup>-3</sup>
4.6 <sup>+</sup>	15	D <sub>1</sub> , $\dot{D}_1$	6.867 × 10 <sup>-1</sup>	4.414 × 10 <sup>-1</sup>	5.904 × 10 <sup>-1</sup>	9.533 × 10 <sup>-6</sup>	8.914 × 10 <sup>-6</sup>	1.309 × 10 <sup>-5</sup>
5.0 <sup>+</sup>	G <sub>1</sub>	----	8.867 × 10 <sup>-1</sup>	7.417 × 10 <sup>-1</sup>	1.333	4.719 × 10 <sup>-6</sup>	4.543 × 10 <sup>-6</sup>	8.457 × 10 <sup>-6</sup>
5.3 <sup>+</sup>	16	$\dot{D}_1$	8.156 × 10 <sup>-1</sup>	8.735 × 10 <sup>-1</sup>	1.573	1.611 × 10 <sup>-3</sup>	4.843 × 10 <sup>-3</sup>	4.040 × 10 <sup>-3</sup>
6.3 <sup>+</sup>	17	$\dot{D}_1$	1.374	3.007	3.164	6.892 × 10 <sup>-4</sup>	3.039 × 10 <sup>-3</sup>	2.103 × 10 <sup>-3</sup>
10.3 <sup>+</sup>	21	$\dot{D}_1$	9.845	3.961 × 10 <sup>1</sup>	2.885 × 10 <sup>1</sup>	6.835 × 10 <sup>-4</sup>	3.040 × 10 <sup>-3</sup>	2.100 × 10 <sup>-3</sup>
20.3 <sup>+</sup>	26	$\dot{D}_1$	1.375 × 10 <sup>2</sup>	6.291 × 10 <sup>2</sup>	4.433 × 10 <sup>2</sup>	6.537 × 10 <sup>-4</sup>	3.008 × 10 <sup>-3</sup>	2.081 × 10 <sup>-3</sup>
30.0 <sup>+</sup>	G <sub>2</sub>	----	6.664 × 10 <sup>2</sup>	3.637 × 10 <sup>3</sup>	3.397 × 10 <sup>3</sup>	4.329 × 10 <sup>-4</sup>	2.146 × 10 <sup>-3</sup>	1.856 × 10 <sup>-3</sup>
40.3 <sup>+</sup>	36	$\dot{D}_1$	5.355 × 10 <sup>2</sup>	3.633 × 10 <sup>3</sup>	7.359 × 10 <sup>3</sup>	4.008 × 10 <sup>-4</sup>	8.254 × 10 <sup>-4</sup>	1.491 × 10 <sup>-3</sup>
133.5 <sup>+</sup>	46	$\dot{D}_1$	6.173 × 10 <sup>2</sup>	4.982 × 10 <sup>3</sup>	1.323 × 10 <sup>4</sup>	5.114 × 10 <sup>-5</sup>	5.149 × 10 <sup>-4</sup>	1.151 × 10 <sup>-3</sup>
			1.607 × 10 <sup>2</sup>	5.414 × 10 <sup>2</sup>	2.659 × 10 <sup>3</sup>	6.892 × 10 <sup>-6</sup>	1.104 × 10 <sup>-6</sup>	6.198 × 10 <sup>-7</sup>

TABLE 38. - LATE NAVIGATION UNCERTAINTIES FOR AUGMENTED RUN

Data Trajectory time, days	Meas no.	Kind	Position uncertainties, km <sup>2</sup>			Velocity uncertainties, m <sup>2</sup> /sec <sup>2</sup>		
			$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$	$\sigma_X^2$	$\sigma_Y^2$	$\sigma_Z^2$
193.5 <sup>+</sup>	52	$\dot{D}_1$	4.304 x 10 <sup>2</sup>	2.205 x 10 <sup>2</sup>	1.592 x 10 <sup>3</sup>	3.619 x 10 <sup>-6</sup>	5.665 x 10 <sup>-6</sup>	1.265 x 10 <sup>-5</sup>
199.5 <sup>+</sup>	55	$\dot{D}_1$	4.635 x 10 <sup>2</sup>	1.922 x 10 <sup>2</sup>	1.423 x 10 <sup>3</sup>	3.006 x 10 <sup>-6</sup>	5.796 x 10 <sup>-6</sup>	1.392 x 10 <sup>-5</sup>
203.5 <sup>+</sup>	57	$\dot{D}_1$	4.847 x 10 <sup>2</sup>	1.768 x 10 <sup>2</sup>	1.323 x 10 <sup>3</sup>	3.135 x 10 <sup>-6</sup>	6.237 x 10 <sup>-6</sup>	1.502 x 10 <sup>-5</sup>
704.6 <sup>+</sup>	58	$\dot{D}_1$	4.901 x 10 <sup>2</sup>	1.689 x 10 <sup>2</sup>	1.289 x 10 <sup>3</sup>	3.564 x 10 <sup>-6</sup>	2.062 x 10 <sup>-5</sup>	1.559 x 10 <sup>-5</sup>
205.4 <sup>+</sup>	59	Angles	2.609 x 10 <sup>2</sup>	9.938 x 10 <sup>1</sup>	5.927 x 10 <sup>2</sup>	2.762 x 10 <sup>-6</sup>	1.685 x 10 <sup>-5</sup>	1.182 x 10 <sup>-5</sup>
205.6 <sup>+</sup>	60	$\dot{D}_1$	2.609 x 10 <sup>2</sup>	7.596 x 10 <sup>1</sup>	4.918 x 10 <sup>2</sup>	5.084 x 10 <sup>-6</sup>	1.316 x 10 <sup>-5</sup>	1.287 x 10 <sup>-5</sup>
205.8 <sup>+</sup>	61	$\beta$	2.608 x 10 <sup>2</sup>	7.571 x 10 <sup>1</sup>	4.892 x 10 <sup>2</sup>	1.248 x 10 <sup>-5</sup>	1.464 x 10 <sup>-5</sup>	2.014 x 10 <sup>-5</sup>
206.4 <sup>+</sup>	62	Angles	7.740 x 10 <sup>1</sup>	4.680 x 10 <sup>1</sup>	7.223 x 10 <sup>1</sup>	9.358 x 10 <sup>-4</sup>	9.054 x 10 <sup>-4</sup>	3.539 x 10 <sup>-4</sup>
206.6 <sup>+</sup>	63	$\dot{D}_1$	2.611 x 10 <sup>1</sup>	8.240	3.517 x 10 <sup>1</sup>	5.851 x 10 <sup>-1</sup>	3.002 x 10 <sup>-1</sup>	1.875
206.8 <sup>+</sup>	64	$\beta$	1.105 x 10 <sup>3</sup>	1.610 x 10 <sup>2</sup>	8.718 x 10 <sup>2</sup>	3.463	5.856 x 10 <sup>-1</sup>	2.033
207.6 <sup>+</sup>	65	$\dot{D}_1$	7.340 x 10 <sup>3</sup>	3.983 x 10 <sup>3</sup>	6.205 x 10 <sup>3</sup>	9.664 x 10 <sup>-1</sup>	5.204 x 10 <sup>-1</sup>	7.925 x 10 <sup>-1</sup>

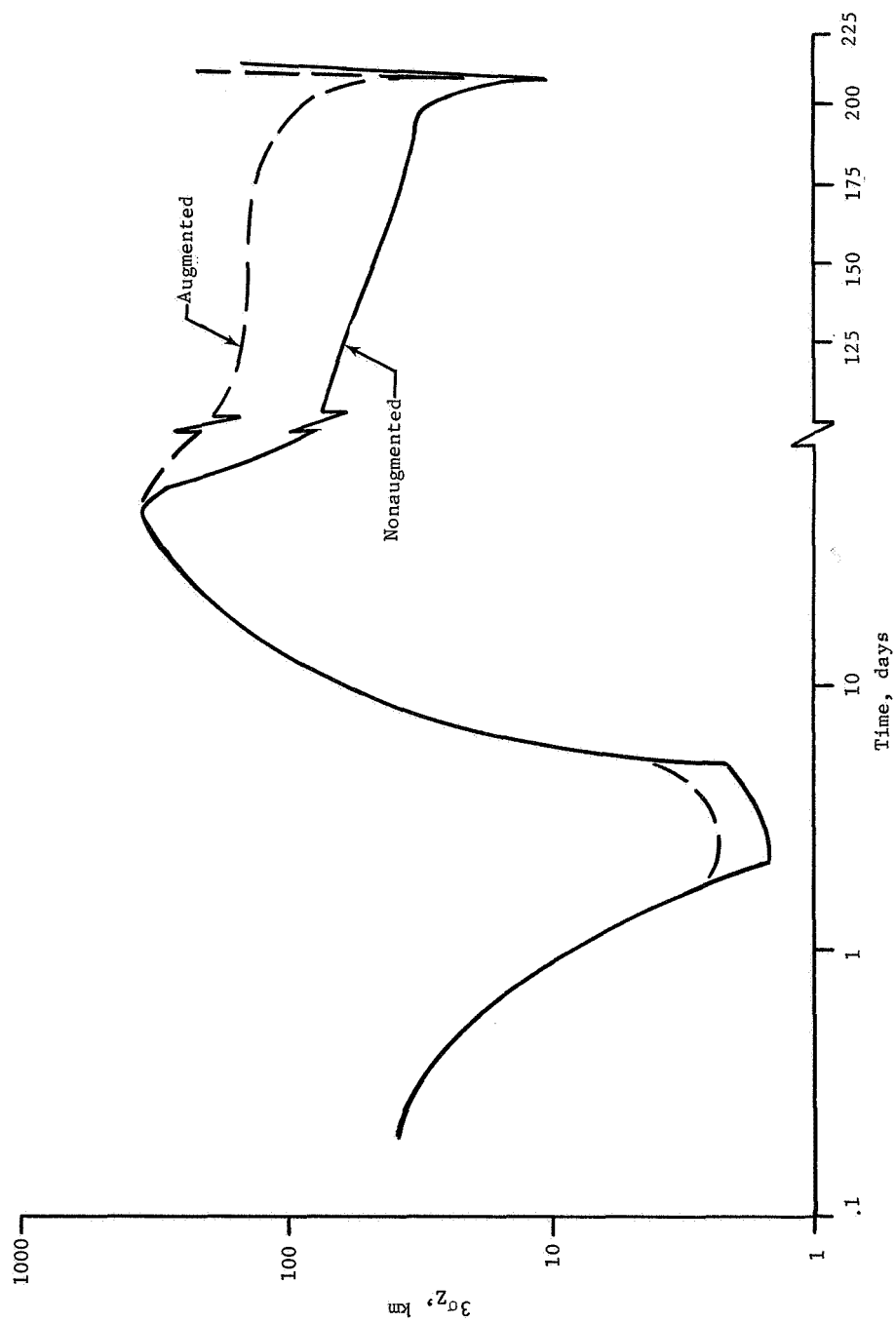


Figure 15.- Navigation Uncertainties for Auxiliary Error Analysis Run



Recall that exercising the error analysis mode with an augmented state vector assumes that the underlying navigation algorithm is operating in a "solve for" mode. Table 39 gives the uncertainties in the various augmented parameters as they are changed during the example trajectory. The initial uncertainties for these parameters are quite small and only two of them, biases in  $\mu$  for both the Sun and the target planet Mars, are significantly altered over the course of the trajectory. The uncertainty in the range bias, for example, is never changed except in the eighth significant figure. Its original  $3\sigma$  uncertainty is only 1.5 m and, effectively, the navigation process never becomes accurate enough to pick up the single contribution of the ranging bias. The  $3\sigma$  initial uncertainty for the geocentric station location bias is only 6 m. Again, position uncertainties of this magnitude are dwarfed by the kilometer magnitude errors in the estimated nonaugmented state vector and no significant reduction results after processing the data.

In other words, the numbers in table 39 indicate that, for these specified uncertainty levels and the given conditions for the example run, the only two uncertainties whose influence can be definitely singled out are the gravitational constant biases for the Sun and the target planet. At the end of the run, the uncertainty in the Sun's gravitational constant has been reduced to a  $3\sigma$  level of  $22.5 \times 10^3 \text{ km}^3/\text{sec}^2$  from an initial value of  $90.0 \times 10^3 \text{ km}^3/\text{sec}^2$ , a reduction by a factor of four. Similarly, the uncertainty in the Mars gravitational constant has come down from  $(3\sigma) \ 9 \text{ km}^3/\text{sec}^2$  to  $3.63 \text{ km}^3/\text{sec}^2$ , about a factor of three.

Results similar to those given in table 39 generally play a key role in setting up an actual orbit determination process for a given mission. The JPL DPODP system, for example, has both a "consider" mode, which reflects uncertainties in various parameters back into the state vector without actually estimating their values, and a "solve for" mode. It would be virtually impossible to run a real-time navigation process with every single parameter imaginable included in the augmented state vector. One can, however, perform a priori studies similar to the one just given and determine, for reasonable uncertainties in all the parameters based on physical knowledge, which parameters should be included in the "solve for" mode and which should just be "considered." The parameters whose uncertainties decrease significantly during the trajectory and bring about reductions in position and velocity errors due to their being estimated should be included in the "solve for" state vector.

TABLE 39.- VARIANCES OF AUGMENTED PARAMETERS FOR AUGMENTED RUN

Data Trajectory Time, Days	Meas No.	Kind	$\sigma_{R_1}^2, m^2$	$\sigma_{\theta_1}^2, mrad^2$	$\sigma_{\phi_1}^2, mrad^2$	$\sigma_{\mu_s}^2, km^6/sec^4$	$\sigma_{\mu_t p}^2, km^6/sec^4$	$\sigma_{D_1}^2, m^2$	$\sigma_{D_1}^2, mm^2/sec^2$
0	0	0	4.000	$1.000 \times 10^{-7}$	$1.000 \times 10^{-7}$	$9.000 \times 10^8$	9.000	$2.500 \times 10^{-1}$	$6.250 \times 10^{-2}$
0.2 <sup>+</sup>	1	$\dot{D}_1$	4.000	$1.000 \times 10^{-7}$	$1.000 \times 10^{-7}$	$9.000 \times 10^8$			$6.250 \times 10^{-2}$
0.6 <sup>+</sup>	3	$\dot{D}_1, \ddot{D}_1$	4.000	$1.000 \times 10^{-7}$	$1.000 \times 10^{-7}$	$9.000 \times 10^8$			$6.224 \times 10^{-2}$
2.6 <sup>+</sup>	9	$\dot{D}_1, \ddot{D}_1$	3.998	$9.997 \times 10^{-8}$	$1.000 \times 10^{-7}$	$8.987 \times 10^8$			$5.883 \times 10^{-2}$
4.6 <sup>+</sup>	15	$\dot{D}_1, \ddot{D}_1$	3.997	$9.996 \times 10^{-8}$	$1.000 \times 10^{-7}$	$8.743 \times 10^8$			$5.662 \times 10^{-2}$
6.3 <sup>+</sup>	17	$\dot{D}_1$	3.997	$9.996 \times 10^{-8}$	$1.000 \times 10^{-7}$	$8.743 \times 10^8$			$5.662 \times 10^{-2}$
10.3 <sup>+</sup>	21	$\dot{D}_1$	3.997	$9.996 \times 10^{-8}$	$1.000 \times 10^{-7}$	$8.742 \times 10^8$			$5.662 \times 10^{-2}$
20.3 <sup>+</sup>	26	$\dot{D}_1$	3.997	$9.994 \times 10^{-8}$	$9.998 \times 10^{-8}$	$8.688 \times 10^8$			$5.662 \times 10^{-2}$
30.3 <sup>+</sup>	31	$\dot{D}_1$	3.996	$9.991 \times 10^{-8}$	$9.996 \times 10^{-8}$	$8.648 \times 10^8$			$5.662 \times 10^{-2}$
40.3 <sup>+</sup>	36	$\dot{D}_1$	3.995	$9.989 \times 10^{-8}$	$9.995 \times 10^{-8}$	$8.647 \times 10^8$			$5.662 \times 10^{-2}$
73.5 <sup>+</sup>	40	$\dot{D}_1$	3.983	$9.952 \times 10^{-8}$	$9.988 \times 10^{-8}$	$8.461 \times 10^8$			$5.661 \times 10^{-2}$
103.5 <sup>+</sup>	43	$\dot{D}_1$	3.979	$9.941 \times 10^{-8}$	$9.987 \times 10^{-8}$	$8.176 \times 10^8$			$5.661 \times 10^{-2}$
133.5 <sup>+</sup>	46	$\dot{D}_1$	3.979	$9.940 \times 10^{-8}$	$9.984 \times 10^{-8}$	$8.008 \times 10^8$			$5.661 \times 10^{-2}$
163.5 <sup>+</sup>	49	$\dot{D}_1$	3.979	$9.939 \times 10^{-8}$	$9.981 \times 10^{-8}$	$7.745 \times 10^8$			$5.661 \times 10^{-2}$
193.5 <sup>+</sup>	52	$\dot{D}_1$	3.979	$9.939 \times 10^{-8}$	$9.981 \times 10^{-8}$	$7.745 \times 10^8$	9.000		$5.661 \times 10^{-2}$
197.5 <sup>+</sup>	54	$\dot{D}_1$	3.978	$9.938 \times 10^{-8}$	$9.976 \times 10^{-8}$	$7.216 \times 10^8$	8.991		$5.660 \times 10^{-2}$
199.5 <sup>+</sup>	55	$\dot{D}_1$	3.978	$9.938 \times 10^{-8}$	$9.976 \times 10^{-8}$	$7.160 \times 10^8$	8.989		$5.659 \times 10^{-2}$
201.5 <sup>+</sup>	56	$\dot{D}_1$	3.978	$9.938 \times 10^{-8}$	$9.976 \times 10^{-8}$	$7.113 \times 10^8$	8.949		$5.659 \times 10^{-2}$
203.5 <sup>+</sup>	57	$\dot{D}_1$	3.978	$9.938 \times 10^{-8}$	$9.975 \times 10^{-8}$	$7.071 \times 10^8$	8.651		$5.659 \times 10^{-2}$
204.6 <sup>+</sup>	58	$\dot{D}_1$	3.978	$9.938 \times 10^{-8}$	$9.975 \times 10^{-8}$	$7.053 \times 10^8$	1.979		$5.659 \times 10^{-2}$
205.4 <sup>+</sup>	59	Angles	3.977	$9.935 \times 10^{-8}$	$9.968 \times 10^{-8}$	$3.492 \times 10^8$	1.978		$5.619 \times 10^{-2}$
205.6 <sup>+</sup>	60	$\dot{D}_1$	3.967	$9.904 \times 10^{-8}$	$9.965 \times 10^{-8}$	$3.042 \times 10^8$	1.591		$5.581 \times 10^{-2}$
206.4 <sup>+</sup>	62	Angles	3.949	$9.851 \times 10^{-8}$	$9.944 \times 10^{-8}$	$9.154 \times 10^7$	1.564		$5.509 \times 10^{-2}$
206.6 <sup>+</sup>	63	$\dot{D}_1$	3.945	$9.838 \times 10^{-8}$	$9.944 \times 10^{-8}$	$6.706 \times 10^7$	1.462		$4.612 \times 10^{-2}$
206.8 <sup>+</sup>	64	$\beta$	3.945	$9.838 \times 10^{-8}$	$9.944 \times 10^{-8}$	$6.705 \times 10^7$	1.462		$4.612 \times 10^{-2}$
207.6 <sup>+</sup>	65	$\dot{D}_1$	3.926	$9.782 \times 10^{-8}$	$9.930 \times 10^{-8}$	$5.643 \times 10^7$	1.459	$2.500 \times 10^{-1}$	$4.599 \times 10^{-2}$

3. Mars in-orbit run.- It is well-known that one of the most difficult current orbit determination problems is the precise computation of vehicle trajectories around another gravitational body using only Earth-based tracking. Some of the orbit determination problems associated with the lunar flights emphasize this difficulty and it should only be expected, given the mathematics of the process, that the problems, will be accentuated for a Mars Orbiter. Thus, even though STEAP is not currently set up in the most advantageous way to handle orbits around a planetary governing body, an error analysis run for a vehicle in orbit around Mars was made to see if anything could be learned.

The conditions for the test run are given in table 40. The nominal orbit is near Mars-synchronous with an inclination of  $40^\circ$  and periapsis-apoapsis parameters of 4800 and 35 300 km, respectively. Since it is intended to place the Viking Orbiter in a similar orbit, such a problem is realistic. Fifty measurements a day for about two orbits were taken from Earth-based tracking stations and the  $3\sigma$  range rate uncertainties were assumed to be 3 mm/sec. The initial position and velocity errors, large because they would result from an orbit injection maneuver of roughly 1300 m/sec on a real flight, are given by ( $3\sigma$ ) 150 km and 18 m/sec.

Table 41 presents the resulting diagonal elements of the processed covariance matrix for this example as well as the distance from the center of Mars and velocity magnitude in areocentric coordinates for selected orbit epochs. Figure 16 is a graph of the same data. The general trend is obvious. Starting at or near periapsis for the orbit, the position uncertainties increase slightly during the first fourth of the orbit while the velocity uncertainties are being reduced by the filtering algorithm. At apoapsis the position errors are still larger than they were initially, while the velocity errors have been reduced roughly an order of magnitude. As the spacecraft heads toward periapsis, the position uncertainties rapidly decrease until after one Mars orbit their  $3\sigma$  values are 4.26, 3.24, and 3.90 km, a reduction by a factor of 35 to 45 for one orbit. The  $3\sigma$  velocity errors after one orbit are down to 1.56, 1.14, and 0.59 m/sec, also a significant reduction.

TABLE 40.- CONDITIONS FOR ERROR ANALYSIS RUN IN MARS ORBIT

Initial date: Feb. 16, 0 hr, 2 min, 20.288 sec, 1974	
Initial conditions: (areocentric ecliptic coordinates)	
X = 2643.62 km	$\dot{X} = 2.286035$ km/sec
Y = -887.45 km	$\dot{Y} = 3.138229$ km/sec
Z = -3898.43 km	$\dot{Z} = 0.83594$ km/sec
$r_M = 4793.13$ km	$V_M = 3.97155$ km/sec
Ephemeris included Sun, Earth, Mars	
No dynamic noise or guidance corrections	
State transition matrices from patched conic.	
Measurement accuracy for all range rate $\sigma_D^2 = 9 \text{ mm}^2/\text{sec}^2$	
Initial covariance diagonal - all positions, $\sigma_P^2 = 2500 \text{ km}^2$	
all velocities, $\sigma_V^2 = 36 \text{ m}^2/\text{sec}^2$	
Measurement schedule - Goldstone range rate 17 times every 0.2 days from 0 to 0.34	
Canberra range rate 17 times every 0.2 days from 0.36 to 0.68	
Madrid range rate 16 times every 0.2 days from 0.70 to 1.00	
Same schedule repeated second day.	

TABLE 41.- NAVIGATION UNCERTAINTIES FOR ERROR ANALYSIS RUN IN MARS ORBIT

Data	Meas No.	Kind	$r_M$ , km	$V_M$ , km/sec	$\sigma_X^2$ , km <sup>2</sup>	$\sigma_Y^2$ , km <sup>2</sup>	$\sigma_Z^2$ , m <sup>2</sup> /sec <sup>2</sup>	$\sigma_X^2$ , m <sup>2</sup> /sec <sup>2</sup>	$\sigma_Y^2$ , m <sup>2</sup> /sec <sup>2</sup>	$\sigma_Z^2$ , m <sup>2</sup> /sec <sup>2</sup>
Trajectory Time, Days										
0	0		4 793	3.9715	2.500 x 10 <sup>3</sup>	2.500 x 10 <sup>3</sup>	2.500 x 10 <sup>3</sup>	3.600 x 10 <sup>1</sup>	3.600 x 10 <sup>1</sup>	3.600 x 10 <sup>1</sup>
0.1 <sup>+</sup>	5	$\dot{D}_1$	17 118	1.6959	5.707 x 10 <sup>3</sup>	3.039 x 10 <sup>3</sup>	2.585 x 10 <sup>3</sup>	1.061 x 10 <sup>2</sup>	6.082 x 10 <sup>1</sup>	7.428 x 10 <sup>1</sup>
0.2 <sup>+</sup>	10	$\dot{D}_1$	25 896	1.0842	2.001 x 10 <sup>4</sup>	1.103 x 10 <sup>4</sup>	1.188 x 10 <sup>4</sup>	3.620 x 10 <sup>1</sup>	2.070 x 10 <sup>1</sup>	3.142 x 10 <sup>1</sup>
0.3 <sup>+</sup>	15	$\dot{D}_1$	31 335	0.7743	2.136 x 10 <sup>4</sup>	1.198 x 10 <sup>4</sup>	1.413 x 10 <sup>4</sup>	6.888	3.926	7.927
0.4 <sup>+</sup>	20	$\dot{D}_3$	34 350	0.5990	1.663 x 10 <sup>4</sup>	9.500 x 10 <sup>3</sup>	1.151 x 10 <sup>4</sup>	3.410 x 10 <sup>1</sup>	1.882 x 10 <sup>1</sup>	8.385 x 10 <sup>1</sup>
0.5 <sup>+</sup>	25	$\dot{D}_3$	35 295	0.5400	1.652 x 10 <sup>4</sup>	9.429 x 10 <sup>3</sup>	1.220 x 10 <sup>4</sup>	3.153 x 10 <sup>1</sup>	1.849 x 10 <sup>1</sup>	3.821 x 10 <sup>3</sup>
0.6 <sup>+</sup>	30	$\dot{D}_3$	34 267	0.6036	1.278 x 10 <sup>4</sup>	7.290 x 10 <sup>3</sup>	1.012 x 10 <sup>4</sup>	2.489	1.417	8.627 x 10 <sup>1</sup>
0.7 <sup>+</sup>	35	$\dot{D}_2$	31 160	0.7835	3.504 x 10 <sup>3</sup>	2.001 x 10 <sup>3</sup>	3.020 x 10 <sup>3</sup>	2.799	1.577	1.421
0.8 <sup>+</sup>	40	$\dot{D}_2$	25 605	1.1006	3.388 x 10 <sup>2</sup>	1.939 x 10 <sup>2</sup>	3.316 x 10 <sup>2</sup>	1.195	6.690 x 10 <sup>1</sup>	7.429 x 10 <sup>1</sup>
0.9 <sup>+</sup>	45	$\dot{D}_2$	16 649	1.7363	1.376 x 10 <sup>1</sup>	7.962	1.971 x 10 <sup>1</sup>	6.377 x 10 <sup>1</sup>	3.564 x 10 <sup>1</sup>	4.872 x 10 <sup>1</sup>
0.96 <sup>+</sup>	48	$\dot{D}_2$	8 757	2.7680	4.445 x 10 <sup>1</sup>	2.285 x 10 <sup>1</sup>	5.156 x 10 <sup>2</sup>	6.047 x 10 <sup>1</sup>	3.421 x 10 <sup>1</sup>	6.341 x 10 <sup>1</sup>
1.0 <sup>+</sup>	50	$\dot{D}_2$	4 884	3.9292	2.093	1.189	1.698	2.789 x 10 <sup>1</sup>	1.488 x 10 <sup>1</sup>	3.935 x 10 <sup>2</sup>
1.04 <sup>+</sup>	52	$\dot{D}_1$	9 990	2.5408	2.543 x 10 <sup>-1</sup>	1.409 x 10 <sup>-1</sup>	4.044 x 10 <sup>-2</sup>	8.599 x 10 <sup>-2</sup>	4.807 x 10 <sup>-2</sup>	4.843 x 10 <sup>-2</sup>
1.1 <sup>+</sup>	55	$\dot{D}_1$	17 577	1.6569	2.006	1.116	9.202 x 10 <sup>-1</sup>	2.546 x 10 <sup>-2</sup>	1.430 x 10 <sup>-2</sup>	1.749 x 10 <sup>-2</sup>
1.2 <sup>+</sup>	60	$\dot{D}_1$	26 183	1.0674	4.483	2.499	2.569	6.048 x 10 <sup>-3</sup>	3.406 x 10 <sup>-3</sup>	5.012 x 10 <sup>-3</sup>
1.3 <sup>+</sup>	65	$\dot{D}_1$	31 508	0.7646	5.884	3.285	3.723	1.457 x 10 <sup>-3</sup>	8.246 x 10 <sup>-4</sup>	1.542 x 10 <sup>-3</sup>
1.4 <sup>+</sup>	70	$\dot{D}_3$	34 431	0.5942	6.426	3.593	4.365	1.364 x 10 <sup>-4</sup>	7.912 x 10 <sup>-5</sup>	3.054 x 10 <sup>-4</sup>
1.5 <sup>+</sup>	75	$\dot{D}_3$	35 293	0.5402	6.257	3.502	4.529	1.094 x 10 <sup>-4</sup>	5.984 x 10 <sup>-5</sup>	2.831 x 10 <sup>-6</sup>
1.6 <sup>+</sup>	80	$\dot{D}_3$	34 182	0.6088	5.371	3.010	4.155	9.949 x 10 <sup>-4</sup>	5.511 x 10 <sup>-4</sup>	3.731 x 10 <sup>-4</sup>
1.7 <sup>+</sup>	85	$\dot{D}_2$	30 982	0.7936	3.699	2.076	3.107	2.862 x 10 <sup>-3</sup>	1.589 x 10 <sup>-3</sup>	1.523 x 10 <sup>-3</sup>
1.8 <sup>+</sup>	90	$\dot{D}_2$	25 310	1.1182	1.607	9.049 x 10 <sup>-1</sup>	1.543	5.706 x 10 <sup>-3</sup>	3.166 x 10 <sup>-3</sup>	3.648 x 10 <sup>-3</sup>
1.9 <sup>+</sup>	95	$\dot{D}_2$	16 173	1.7794	1.795 x 10 <sup>-1</sup>	1.038 x 10 <sup>-1</sup>	2.679 x 10 <sup>-1</sup>	9.891 x 10 <sup>-3</sup>	5.444 x 10 <sup>-3</sup>	7.646 x 10 <sup>-3</sup>
1.96 <sup>+</sup>	98	$\dot{D}_2$	8 138	2.8897	1.891 x 10 <sup>-2</sup>	9.056 x 10 <sup>-3</sup>	6.272 x 10 <sup>-4</sup>	1.116 x 10 <sup>-2</sup>	6.018 x 10 <sup>-3</sup>	1.228 x 10 <sup>-2</sup>
2.00 <sup>+</sup>	100	$\dot{D}_2$	5 145	3.8143	4.299 x 10 <sup>-2</sup>	2.275 x 10 <sup>-2</sup>	4.246 x 10 <sup>-2</sup>	1.349 x 10 <sup>-2</sup>	6.978 x 10 <sup>-3</sup>	3.116 x 10 <sup>-3</sup>

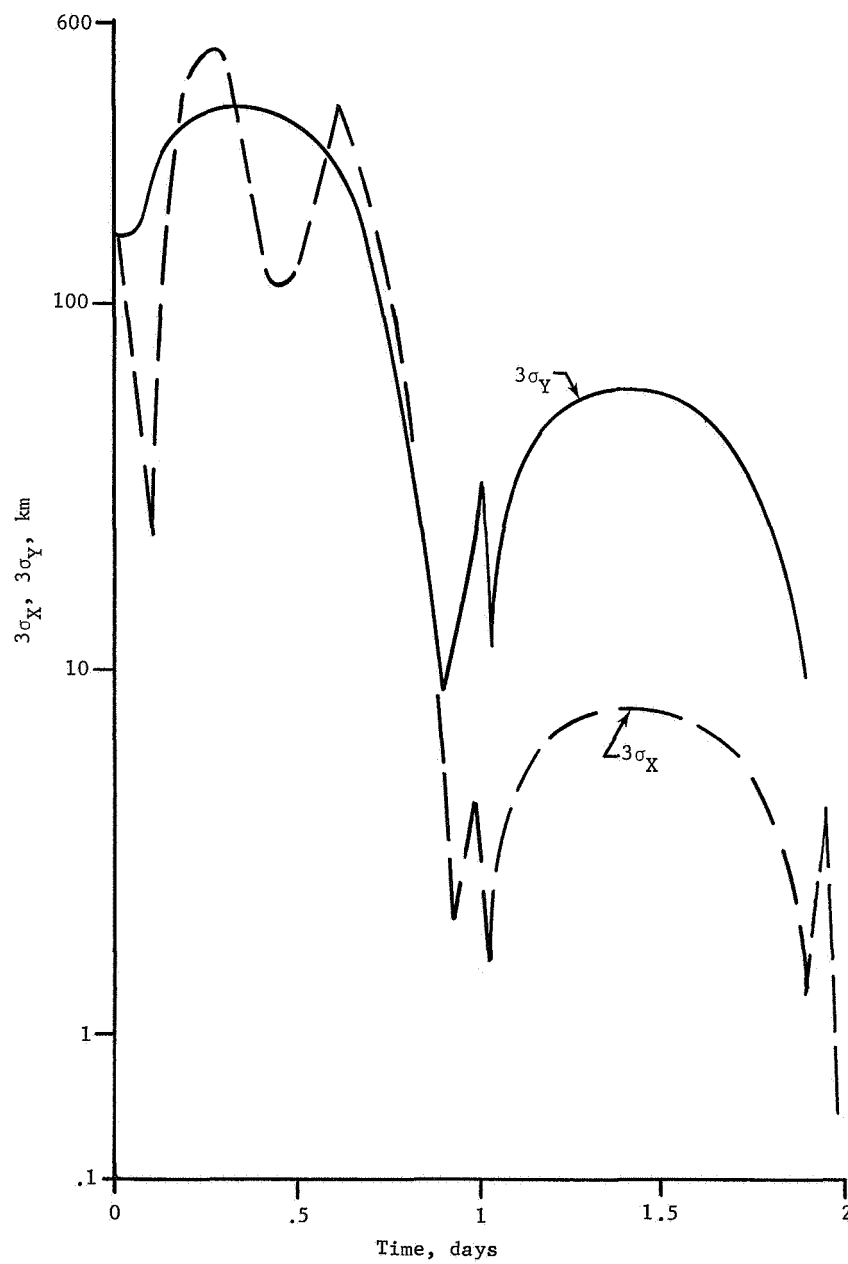


Figure 16.- Navigation Uncertainties for Error Analysis Run in Mars Orbit

On the second orbit the process essentially repeats the general trend. The position errors increase while the vehicle moves away from Mars and the estimation algorithm continues to filter effectively to reduce the velocity uncertainties. After approximately two orbits the  $3\sigma$  position errors are down to 0.63, 0.45, and 0.63 km, an additional reduction by a factor of 5 to 8 for the second orbit. The  $3\sigma$  velocity uncertainties after two orbits have been reduced to 0.345, 0.251, and 0.168 m/sec. Thus it appears, on the surface, that a Mars orbiter can be very successfully tracked from Earth-based radar stations.

Table 42 adequately demonstrates why precise orbit computation for a problem such as the one postulated is very difficult. Recall that a dynamic system's nonobservability, for a recursive navigation process, manifests itself in terms of correlation coefficients near unit magnitude. The correlated uncertainties of the state vector elements throw some doubt on the usual interpretation of table 42. Errors in every single component of the state vector are highly correlated with errors in every other component. If the system is observable at all (that is, has a unique solution), it is highly ill-conditioned.

The meaning of all the nines in the correlation coefficient table is that the individual error components are just barely separable under the set of assumed observations and data processing schemes. More importantly, and herein lies the primary problem in determining a spacecraft's orbit around another governing body from Earth-based tracking, this ill-conditioning means that the orbit estimate is extremely sensitive to slight changes in the measurements. Since these slight changes in the measurements could be caused by dynamic forces not modelled in the equations, one is led to the natural conclusion that the orbit estimate is highly unstable and that, even if there is theoretically a unique solution for the orbit, imprecise knowledge of all the governing influences make its realization unlikely. In a sense, then, the variances given in table 41 are worthless. Only if everything about the spacecraft's orbit and the observations is known perfectly do these position and velocity uncertainties have any meaning.

The only way to test the reaction of an orbit determination process with cross-correlations in errors like those in table 42 is to run many simulation mode studies. Preliminary investigations have been made that confirm the inherent instability of the estimation algorithm. Additional buttressing of STEAP to handle in-orbit problems will be required before these results can be published and any conclusions substantiated.

<sup>b</sup> Indicates number would round off to unit magnitude with five figures.



4. Venus run.- Table 43 presents the program input used for a sequence of Venus runs from the proposed Planetary Explorer launch window. The injection date is March 9, 1972, and the date of closest approach to Venus is September 6, 1972. The targeting mode of STEAP was again used to yield injection conditions consistent with a Cape Kennedy launch, a nominal circular parking orbit, and Venus sphere of influence conditions  $B \cdot T = 13\,295$  km with  $B \cdot R = 8484$  km. Guidance corrections were made at variable times using the fuel-saving two-variable B-plane policy. The measurement schedule and accuracies are given in the table as well as the assumed execution accuracy for all runs. The injection covariance matrix was computed from  $V$ ,  $\Sigma$ ,  $\gamma$ , errors for the assumed Planetary Explorer launch vehicle.

Six Venus runs were made with the error analysis mode of STEAP, all exactly the same in every respect except for the times of the guidance corrections. The purpose of the study was to analyze the effect of various midcourse correction times on the key mission analysis parameters computed by the error analysis mode.

Table 44 presents a summary of the  $\Delta V$  information for the six different runs. The first row on the table tells the times of the assumed guidance maneuvers for each of the runs. The most striking quality of table 44 is that the magnitudes of the expected  $\Delta V$  maneuvers are not significantly influenced by varying the correction times. This result has been noticed in almost all test runs for single leg missions. The last row on the table gives a total sum of most likely magnitudes plus  $3\sigma$  values and is probably a good indication of the fuel loading requirements. The savings resulting from performing four corrections probably does not justify the additional sequencing and manpower required at each correction time.

Another fact that stands out from the data presented in table 44 is that it is the first correction that requires the bulk of the fuel available for the midcourse maneuvers. In a sense, since subsequent corrections are relatively small, one could say that for the numbers used in these example Venus runs, the midcourse execution process is much more accurate than the launch vehicle being employed. An a priori optimization, in terms of fuel expenditure, would probably show that only a partial correction (that is, only partially nulling predicted target condition deviations with the midcourse maneuver) at the first maneuver would lessen the fuel-loading requirements. However, the current error analysis mode of STEAP does not allow for such partial corrections.

TABLE 43.- CONDITIONS FOR VENUS ERROR ANALYSIS RUN

Injection date: Mar. 9, 22 hr, 36 min, 46.782 sec, 1972

Closest approach date: Sept. 6, 8 hr, 59 min, 29.518 sec, 1972

Injection conditions: (geocentric ecliptic coordinates)

X = 4814.86 km	$\dot{X} = 7.482996$ km/sec
Y = -3369.75 km	$\dot{Y} = 8.603163$ km/sec
Z = -2931.37 km	$\dot{Z} = 1.520052$ km/sec
$r_E = 6567.42$ km	$V_E = 11.503052$ km/sec

Venus sphere of influence conditions:

B = 15 856 km	B • R = -8 484 km
B • T = -13 395 km	$t_{SI} =$ Sept. 4, 23 hr, 59 min, 53.967 sec, 1972

Ephemeris included Earth, Venus, Sun, Jupiter, Moon

Accuracy level:  $5 \times 10^{-6}$  True anomaly: 7.696 mrad

No dynamic noise

State transition matrices from patched conic

Guidance corrections at variable times, 2-variable B-plane policy

Execution errors:  $\sigma_{pro}^2 = 1 \times 10^{-4}$   $\sigma_{res}^2 = 100$  cm<sup>2</sup>/sec<sup>2</sup>

$\sigma_{\delta\alpha}^2 = \sigma_{\delta\beta}^2 = 4 \times 10^{-5}$  rad<sup>2</sup>

Measurement schedule - Goldstone measured range and range rate 5 times at 0.2, 1.2, 2.2, 3.2, 4.2, Madrid measured range and range rate 5 times at 0.6, 1.6, 2.6, 3.6, 4.6 Canberra measured range and range rate 5 times at 0.9, 1.9, 2.9, 3.9, 4.9 Goldstone measured range rate 22 times at 5.3, 7.3, 9.3, 11.3, 13.3, 15.3, 60.4, 75.4, 90.4, 105.4, 120.4, 135.4, 150.4, 165.2, 167.2, 171.2, 173.2, 175.2, 177.2, 179.2, 181.2 Madrid measured range rate 13 times at 6.6, 8.6, 10.6, 12.6, 14.6, 16.6, 65.7, 80.7, 95.7, 110.7, 125.7, 140.7, 155.7 Canberra measured range rate 24 times at 17.9, 22.9, 27.9, 32.9, 37.9, 42.9, 47.9, 52.9, 57.9, 70.8, 85.8, 100.8, 115.8, 130.8, 145.8, 160.8, 166.9, 168.9, 170.9, 172.9, 174.9, 176.9, 178.9, 180.9

Total of 74 measurement times

Measurement accuracies:  $\sigma_D^2 = 225$  m<sup>2</sup> for all ranges  
 $\sigma_{\dot{D}}^2 = 9$  mm<sup>2</sup>/sec<sup>2</sup> for all range rates

Injection covariance computed from V,  $\gamma$ ,  $\Sigma$  uncertainties

$P_0 =$	<table><tbody><tr><td>9.000</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>9.000</td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>9.000</td><td></td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>0</td><td><math>1.4762 \times 10^{-3}</math></td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>0</td><td><math>-1.3219 \times 10^{-3}</math></td><td><math>2.4753 \times 10^{-3}</math></td><td></td></tr><tr><td>0</td><td>0</td><td>0</td><td><math>-1.769 \times 10^{-4}</math></td><td><math>-1.6308 \times 10^{-3}</math></td><td><math>2.8009 \times 10^{-3}</math></td></tr></tbody></table>	9.000						0	9.000					0	0	9.000				0	0	0	$1.4762 \times 10^{-3}$			0	0	0	$-1.3219 \times 10^{-3}$	$2.4753 \times 10^{-3}$		0	0	0	$-1.769 \times 10^{-4}$	$-1.6308 \times 10^{-3}$	$2.8009 \times 10^{-3}$
9.000																																					
0	9.000																																				
0	0	9.000																																			
0	0	0	$1.4762 \times 10^{-3}$																																		
0	0	0	$-1.3219 \times 10^{-3}$	$2.4753 \times 10^{-3}$																																	
0	0	0	$-1.769 \times 10^{-4}$	$-1.6308 \times 10^{-3}$	$2.8009 \times 10^{-3}$																																

Units are km and km<sup>2</sup>/sec<sup>2</sup>

TABLE 44.- COMPARISON OF VENUS RUNS,  $\Delta V$ 

Data	Run number					
	1	2	3	4	5	6
Corrections	5, 80, 155	5, 40, 155	5, 80, 170	7.5, 80, 155	3, 8, 80, 155	3, 120, 170
$E[ \Delta V_1 ]$ , m/sec	67.457	67.457	67.457	68.573	66.353	66.353
$\sigma \Delta V_1 $ , m/sec	45.739	45.739	45.739	46.533	44.966	44.966
$E[ \Delta V_2 ]$ , m/sec	1.835	0.945	1.835	1.806	0.695	3.745
$\sigma \Delta V_2 $ , m/sec	1.342	0.592	1.342	1.316	0.369	2.802
$E[ \Delta V_3 ]$ m/sec	0.731	1.181	1.909	0.727	0.254	0.231
$\sigma \Delta V_3 $ , m/sec	0.551	0.889	1.441	0.549	0.191	0.152
$E[ \Delta V_4 ]$ , m/sec	0	0	0	0	0.701	0
$\sigma \Delta V_4 $ , m/sec	0	0	0	0	0.530	0
$\sum E[ \Delta V_i ]$ , m/sec	70.023	69.583	71.201	71.106	68.003	70.329
$\sum E[ \Delta V_i ] + 3\sigma_i$ , m/sec	212.919	211.243	216.767	216.300	206.171	214.089

Table 45 and 46 compare target condition uncertainties after every sequence of one or two corrections considered by the example runs. From the results, it is apparent that whether the first correction is made at 3, 5, or 7.5 days is not important from the point of view of target condition uncertainties. Table 46 demonstrates, as would be expected for this example, that the later the second correction is made, the less the likely errors on encountering the impact parameter plane. This is only natural since the terms of the variation matrix  $\eta$  are going down throughout the flight.

Table 47 presents the final target condition uncertainties (likely deviations from nominal target conditions) as a result of using each of the six guidance correction schedules. The four schedules that have the last maneuver 155 days after injection show markedly similar results. However, one of the schedules that assumes corrections at 5, 40, and 155 days has an uncertainty ellipse semiminor axis that is noticeably smaller. The most interesting features of table 47 are the orientation of the uncertainty ellipses in the B-plane. Depending on the location of the target aim point, errors in B·T and B·R may be translated into errors in hyperbolic inclination and radius of closest approach. The schedules studied show a shift of up to 7° in the orientation of the impact parameter plane uncertainty ellipse. If, for the given mission, uncertainties in inclination could be tolerated more readily than errors in periapsis distance (worry about impacting Venus, for example), then the correction schedules could be chosen to produce a B-plane uncertainty ellipse alignment that reduced likely periapsis distance errors.

Table 48 compares the diagonal elements of the navigation uncertainty covariance matrix at the time of closest approach to Venus. The results might have been anticipated. The schedule employing four corrections has the least navigation uncertainty because its final correction, and hence its execution error, were small in magnitude. The schedules that made the last correction at 170 days have larger navigation uncertainties because their final execution errors have not been completely filtered. Table 48 also indicates that making the corrections at 5, 40, and 155 days results in lower tracking errors at the planet than the other two schedules with three corrections that make the third maneuver at 155 days. A detailed investigation of the available output turned up one of the reasons why midcourse guidance optimization can be important. Between 40 and 80 days, the likely direction of the second correction shifts from a direction of high navigation accuracy to a direction of low navigation accuracy. Thus, a correction at 40 days is most likely to be made in a direction with small navigation uncertainties. As a result, at the time of the last correction, errors in the Z component of position were significantly less if the second correction was made at 40 days.

Midcourse guidance optimization is a subject that has not been pursued sufficiently. For multiplanet missions, its application is essential. And as the above results demonstrate, even for single-leg missions much can be learned by taking a preliminary look at several candidate correction schedules.

TABLE 45.- COMPARISON OF TARGET CONDITION UNCERTAINTIES  
FOR DIFFERENT 1ST MIDCOURSE TIMES

Correction Data	After injection	If 1st midcourse at 3 days	If 1st midcourse at 5 days	If 1st midcourse at 7.5 days
$3\sigma_{B \cdot T}$ , km	$1.1046 \times 10^7$	90 014	90 144	90 091
$3\sigma_{B \cdot R}$ , km	$4.3624 \times 10^6$	35 454	35 965	35 816
$3\sigma_{\max}$ eigenvalue, km	$1.1948 \times 10^7$	96 570	96 877	96 766
$3\sigma_{\min}$ eigenvalue, km	$3.0797 \times 10^5$	5 795	5 883	5 959
Orientation <sup>a</sup> of ellipse in B-plane, deg	338.1	338.1	338.5	338.6
<sup>a</sup> Angle measured is to major axis from T axis (positive is counter-clockwise).				

TABLE 46.- COMPARISON OF TARGET CONDITION UNCERTAINTIES  
AFTER TWO CORRECTIONS

Corrections Data	1st at 3 2nd at 8	1st at 3 2nd at 120	1st at 5 2nd at 40	1st at 5 2nd at 80	1st at 7½ 2nd at 80
$3\sigma_{B \cdot T}$ , km	13 393	718	10 037	6345	6307
$3\sigma_{B \cdot R}$ , km	5 220	247	2 978	1602	1609
$3\sigma_{\max}$ , km	14 371	719	10 459	6543	6507
$3\sigma_{\min}$ , km	279	246	471	152	153
Orientation <sup>a</sup> of ellipse in B-plane, deg	338.7	357.6	343.7	345.9	345.7
<sup>a</sup> Angle measured is to major axis from T axis (positive is counterclockwise).					

TABLE 47.- COMPARISON OF TARGET CONDITION UNCERTAINTIES AFTER ALL CORRECTIONS

Corrections Data	5,80,155	5,40,155	5,80,170	7½,80,155	3,8,80,155	3,120,170
$3\sigma_{B.T}$ , km	653	639	247	652	648	258
$3\sigma_{B.R}$ , km	364	374	162	365	360	159
$3\sigma_{max}$ , km	744	739	285	743	737	277
$3\sigma_{min}$ , km	80	54	78	80	77	123
Orientation <sup>a</sup> of ellipse in B-plane, deg	28.8	30.2	31.3	28.9	28.9	24.0
<sup>a</sup> Angle measured is to major axis from T axis (positive is counterclockwise).						

TABLE 48.- COMPARISON OF NAVIGATION UNCERTAINTIES AT CLOSEST APPROACH

Correction schedule	Data	$\sigma_X^2, \text{km}^2$	$\sigma_Y^2, \text{km}^2$	$\sigma_Z^2, \text{km}^2$	$\sigma_{\dot{X}}^2, \text{m}^2/\text{sec}^2$	$\sigma_{\dot{Y}}^2, \text{m}^2/\text{sec}^2$	$\sigma_{\dot{Z}}^2, \text{m}^2/\text{sec}^2$
5, 80, 155		101.827	153.321	1671.071	5.2895	5.3257	4.4608
5, 40, 155		65.859	93.507	575.390	1.9694	1.7790	1.8345
5, 80, 170		114.261	158.828	1675.279	5.3299	5.2868	4.4959
7½, 80, 155		101.925	160.135	1706.761	5.4294	5.4138	4.5742
3, 8, 80, 155		49.473	68.796	280.125	1.0317	0.8508	1.0363
3, 120, 170		181.458	347.299	4230.541	13.1956	13.6076	11.5673



#### D. Simulation Mode

Results from several runs made with the simulation mode of STEAP will be presented in this section. It is hoped that these few examples will indicate some of the many uses for the simulation mode.

The simulation mode differs considerably from the error analysis mode. Within the simulation mode of STEAP, one portion of the program computes an "actual" trajectory and generates observational data. These data are then fed into the estimation algorithm that attempts to reproduce the "actual" trajectory. The details of this process have been discussed in an earlier section.

Table 49 presents the input values used for example run 1 of the simulation mode. The basic trajectory, which was the same throughout all the simulation mode example runs, is a Mars transfer from the Viking launch window. Again the injection conditions, generated by the targeting mode of STEAP, were chosen to satisfy the Mars sphere of influence conditions  $B \cdot T = 4452$  km,  $B \cdot R = 8548$  km,  $t_{SI} = 204.417$  days after injection. Guidance corrections, using a three-variable B-plane policy, were simulated at 5, 30, and 180 days after injection. The injection covariance matrix was diagonal with the initial  $3\sigma$  position errors given by 3 km and the initial  $3\sigma$  velocity uncertainties specified as 9 m/sec. All of the above data remained constant throughout the example simulation mode runs.

The first run was designed to show a case of strong orbit determination convergence. The only input used that impedes convergence is the calculation of the state transition matrices from the analytic patched conic. This was done to conserve computer time so a nonzero process noise matrix  $Q$ , with all the unmodelled acceleration constants  $k$  equal to  $10^{-22}$  km<sup>2</sup>/sec<sup>4</sup>, was added to offset the inaccuracy of the state transition matrices. The nominal trajectory was updated by quasi-linear filtering events at 2.5, 5, 30, 100, and 180 days; the updating was done after the guidance maneuvers when the two occurred simultaneously. Table 49 shows the measurement schedule for the 71 measurements used by the example. The actual measurement accuracy was an order of magnitude higher than the measurement accuracy assumed by the estimation algorithm; that is, in generating the actual measurements, white noise samples corrupting the measurement were chosen from distributions whose standard deviations were an order of magnitude less than those that were assumed by the navigator.

TABLE 49.- CONDITIONS FOR SIMULATION MODE RUN 1

Injection date: July 24, 9 hr, 25 min, 47.637 sec, 1973

Closest approach date: Feb. 16, 0 hr, 15 min, 42.8 sec, 1974

Injection conditions: (geocentric ecliptic coordinates)

X = -1131.91 km	$\dot{X}$ = 9.35948 km/sec
Y = -6575.61 km	$\dot{Y}$ = -2.88174 km/sec
Z = 75.601 km	$\dot{Z}$ = 6.21518 km/sec
$r_E$ = 6672.75 km	$V_E$ = 11.59883 km/sec

Mars sphere of influence conditions:

B = 9637.64 km	B · R = 8547.70 km
B · T = 4452.06 km	$t_{SI}$ = 204.41694 days after injection

Ephemeris included Earth, Sun, Mars, Jupiter, Moon

Accuracy level:  $5.000 \times 10^{-6}$  True anomaly: 7.696 mrad

Guidance corrections at 5, 30, 180, 3-variable B-plane policy

Quasi-linear filtering events at 2.5, 5, 30, 100, and 180

State transition matrices from patched conic

Injection covariance diagonal: All position uncertainties,  $\sigma_P^2 = 1 \text{ km}^2$

All velocity uncertainties,  $\sigma_V^2 = 9 \text{ m}^2/\text{sec}^2$

Dynamic noise constant:  $k = 10^{-22} \text{ km}^2/\text{sec}^4$

Measurement schedule - Goldstone measured range and range rate 5 times at 0.2, 1.3, 2.2, 3.3, 4.2  
 Madrid measured range and range rate 5 times at 0.6, 1.7, 2.6, 3.7, 4.6  
 Canberra measured range and range rate 5 times at 0.9, 1.0, 2.9, 3.0, 4.9  
 Goldstone measured range rate 25 times at 5.3, 8.3, 11.3, 14.3, 17.3, 20.3,  
 23.3, 26.3, 29.3, 30.3, 53.3, 76.3, 99.3, 122.3, 145.3, 168.3, 191.3, 200.3,  
 201.3, 202.3, 204.3, 205.3, 206.3, 207.3  
 Madrid measured range rate 16 times at 6.6, 9.6, 12.6, 15.6, 18.6, 21.6,  
 24.6, 27.6, 37.6, 60.6, 83.6, 106.6, 129.6, 152.6, 175.6, 198.6  
 Canberra measured range rate 15 times at 7.9, 10.9, 13.9, 16.9, 19.9, 22.9,  
 25.9, 28.9, 44.9, 67.9, 90.9, 113.9, 136.9, 159.9, 182.9

Total of 71 measurement times

Assumed measurement accuracy:  $\sigma_D^2 = 25 \times 10^{-6} \text{ km}^2$  for all ranges

$\sigma_{\dot{D}}^2 = 9 \times 10^{-12} \text{ km}^2/\text{sec}^2$  for all range rates

Actual trajectory ephemeris and accuracy level same as nominal

No actual station location biases, measurement biases, dynamic biases, or unmodeled accelerations

Actual measurement accuracy:  $\sigma_D^2 = 25 \times 10^{-8} \text{ km}^2$  for all ranges

$\sigma_{\dot{D}}^2 = 9 \times 10^{-14} \text{ km}^2/\text{sec}^2$  for all range rates

Assumed execution accuracy:

Actual execution errors:

	Correction	1	2	3
$\sigma_{pro}^2 = 25 \times 10^{-6}$				
$\sigma_{res}^2 = 9 \times 10^{-10} \text{ km}^2/\text{sec}^2$	k	-0.0025	0.004	0.002
$\sigma_{\delta\alpha}^2 = \sigma_{\delta\beta}^2 = 5 \times 10^{-5} \text{ rad}^2$	S	$1.5 \times 10^{-5}$	$-1.8 \times 10^{-5}$	$1.1 \times 10^{-5} \text{ km/sec}$
	$\delta\alpha$	-0.0054	0.002	0.007
	$\delta\beta$	0.006	-0.004	0.002

For example, in run 1 the actual and assumed trajectories were computed from n-body routines using the same ephemeris and step size. In addition, the actual dynamic and measurement biases used in the computation of the actual trajectory and observations were all zero. The assumed and actual execution errors are also given in table 49. To summarize, everything about the problem except the computation of the state transition matrices was chosen to produce convergence.

Table 50 contains a chart of the difference between the estimated and actual trajectories throughout the example run. The initial deviations were chosen to be  $\pm 0.5$ , based on the injection covariance matrix. Thus, the initial errors in the navigation process were half a kilometer for each position component and 1.5 m/sec for each velocity component. The chart shows that after one day of taking measurements the position errors are about 1 km, while the velocity errors, as predicted earlier in an error analysis mode run, are down to less than 10 mm/sec.

Recall that in the simulation mode an actual trajectory is being flown and concurrently estimated. The numbers in table 50 are the differences between the actual trajectory and the estimated trajectory. The estimated trajectory results from a Kalman recursive filtering algorithm that operates on simulated measurements generated within the program. Thus, the example represents a "test" of the underlying orbit determination process.

Between 1.7 days after injection and the first guidance correction at injection plus five days, the orbit determination inaccuracies increase measurably. The cause of the increase is the error in the state transition matrices. During this period the hypothetical spacecraft is in the transition region from the Earth's sphere of influence to the Sun's. Since the state transition matrices are computed in terms of a two-body conic around the governing body, they are determined from either a Sun-based ellipse or an Earth-based hyperbola, thereby neglecting the influence of the other body. At the time of the guidance correction, when the vehicle has essentially entered its heliocentric phase, the position inaccuracies are between 3 and 8 km, while the velocity errors are between 9 and 24 mm/sec.

TABLE 50.- ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 1

Trajectory time, days	Meas no.	$dX - \hat{\delta X}$ , km	$dY - \hat{\delta Y}$ , km	$dZ - \hat{\delta Z}$ , km	$d\dot{X} - \hat{\delta \dot{X}}$ , m/sec	$d\dot{Y} - \hat{\delta \dot{Y}}$ , m/sec	$d\dot{Z} - \hat{\delta \dot{Z}}$ , m/sec
0	0	0.5000	-0.5000	0.5000	-1.5000	+1.5000	-1.5000
0.2	1	0.0446	1.7859	-1.4518	-.0154	.1083	-.0898
0.6	2	2.3158	-2.6568	-1.1971	.0393	-.0324	-.0235
0.9	3	-0.5071	0.7014	-0.1333	-.0188	.0178	.0141
1.0	4	0.9244	-0.4659	-1.4281	-.0028	.0035	-.0002
1.3	5	0.0652	-0.3745	0.0167	-.0147	.0079	.0117
1.7	6	-1.3646	1.7986	-0.3311	-.0292	.0272	.0086
2.2	7	-3.5066	2.7680	2.1457	-.0431	.0339	.0278
2.5	QL <sub>1</sub>	-4.6322	3.6631	2.8696	-.0438	.0351	.0281
2.6	8	-1.7428	5.3820	-4.0696	-.0158	.0386	-.0126
2.9	9	-4.3901	8.8408	-3.6306	-.0278	.0497	-.0075
3.0	10	-2.0215	7.1929	-5.8888	-.0151	.0383	-.0212
3.3	11	-2.6637	8.2996	-6.0889	-.0163	.0389	-.0186
3.7	12	-2.4639	7.0009	-4.8979	-.0158	.0298	-.0126
4.2	13	-2.8979	7.8795	-5.2467	-.0146	.0291	-.0117
4.6	14	-4.1589	8.5228	-3.9148	-.0181	.0282	-.0066
4.9	15	-2.9074	7.5045	-4.8185	-.0126	.0238	-.0088
5.0 <sup>+</sup>	G <sub>1</sub>	3.0173	7.7149	-4.8991	.0020	.0273	-.0475
5.3	16	-2.9031	8.4711	-6.0869	.0044	.0294	-.0461
6.6	17	-2.3546	11.7046	-11.3272	.0047	.0289	-.0464
7.9	18	-0.8910	11.3442	-13.9538	.0079	.0146	-.0362
8.3	19	0.4993	10.7296	-15.5525	.0187	.0107	-.0373
9.6	20	1.4059	11.9215	-18.8473	.0108	.0107	-.0349
10.9	21	3.7276	9.8189	-20.5071	.0128	.0041	-.0305

TABLE 50.- ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 1 - Concluded

Trajectory time, days	Meas no.	$dX - \hat{\delta X}$ , km	$dY - \hat{\delta Y}$ , km	$dZ - \hat{\delta Z}$ , km	$d\dot{X} - \hat{\delta \dot{X}}$ , m/sec	$d\dot{Y} - \hat{\delta \dot{Y}}$ , m/sec	$d\dot{Z} - \hat{\delta \dot{Z}}$ , m/sec
12.6	23	4.2156	8.7838	-19.8098	.0108	.0018	-.0223
14.3	25	6.4860	6.3862	-20.8794	.0115	-.0016	-.0196
16.9	27	7.7134	4.6561	-20.6205	.0103	-.0029	-.0148
18.6	29	6.1877	4.1707	-17.0553	.0078	-.0029	-.0099
20.3	31	5.7858	3.2096	-15.1631	.0064	-.0035	-.0074
23.3	34	6.9922	2.6459	-16.8691	.0060	-.0035	-.0072
26.3	37	8.5614	-1.4578	-14.2703	.0061	-.0053	-.0047
29.3	40	7.7296	-2.0557	-12.0429	.0050	-.0050	-.0030
30.0	$G_2^+$	8.0324	-2.3590	-12.2252	.0013	.0065	-.0161
44.9	43	15.4453	-2.3768	-28.4754	.0044	.0006	-.0108
67.9	46	11.5008	3.7427	-31.8385	.0008	.0009	-.0063
83.6	48	8.0972	4.1296	-29.0395	-.0006	.0007	-.0039
99.3	50	7.5384	-.0171	-28.2674	-.0103	-.0001	-.0240
100.0	$QL_4$	7.4760	-.0207	-28.4111	-.0103	-.0001	-.0236
122.3	53	6.2417	-2.2319	-31.2533	-.0105	-.0002	-.0125
145.3	56	3.0569	2.5000	-33.0843	-.0009	.0009	-.0002
159.9	58	2.9504	0.9619	-33.1943	-.0009	.0003	.0003
175.6	60	1.2047	2.0301	-32.4100	-.0010	.0005	.0008
180.0	$G_3^+$	0.8296	2.2388	-32.0671	-.0086	-.0069	.0041
198.6	63	-0.7183	2.9729	-29.2142	-.0015	.0007	.0014
202.3	66	2.4228	-.0841	-29.2197	-.0005	.0002	.0015
203.3	67	2.2982	.0099	-29.0708	-.0005	.0002	.0017
<sup>a</sup> 204.3	68	1.9931	.2608	-28.8914	-.0005	.0003	.0021
<sup>b</sup> 205.3	69	-0.9607	.1399	7.1488	-.0006	-.0003	.0039
<sup>c</sup> 206.3	70	1.4728	-1.3397	0.3714	-.0010	.0011	.0032
<sup>d</sup> 207.3	71	-17.2326	12.4809	-10.0277	-.2894	.2109	-.1669
<sup>a</sup> Distance from Mars at 204.3 = 595 000 km. <sup>b</sup> Distance from Mars at 205.3 = 343 000 km. <sup>c</sup> Distance from Mars at 206.3 = 89 000 km. <sup>d</sup> Distance from Mars at 207.3 = 182 000 km.							

In a similar error analysis run, propagated covariance matrices between 5 and 30 days after injection indicated that the Z component of position was not as observable as the other two. For the first example run, the Z component of the estimated trajectory's deviation from the actual is indeed the most inaccurate. Also, as was predicted by the error analysis mode, the position inaccuracies decrease significantly as the Martian periapsis is approached. Just after periapsis, however, both the position and velocity errors increase significantly.

Table 51 compares the actual measurement residuals with their a priori statistics. As was pointed out in an earlier discussion of the problem of divergence, it is when there is an obvious inconsistency between these residuals and their statistics that divergence is occurring. Two factors should be remembered with respect to table 51. First, although dynamic noise is added throughout the flight, the only intervals where dynamic modeling insufficiency is actually present is in the two transition regions from Earth to Sun domination and later from Sun to Mars domination. The error in these transition regions, as previously noted, is caused by improper state transition matrices. Secondly, the actual measurement white noise is an order of magnitude less than the assumed noise, so if no dynamic noise or nonlinearities of any kind influenced the trajectory, the measurement residuals should look as if they come from a distribution with an even smaller standard deviation than the one given by the a priori statistics.

Table 51 presents three columns of data. The first column is the actual range rate residual; that is, the difference between the predicted observation and the simulated observation. The second column is the a priori statistical variance of the residual computed, as described in an earlier section, from  $\text{HPH}^T + \text{R}$ . The third column results from dividing the square root of a priori range rate variance into the actual measurement residual. To check for statistical consistency, one can compare the values  $k$  with likely random samples taken from a Gaussian distribution with mean zero and unit variance.

TABLE 51.- STATISTICAL CONSISTENCY RESIDUAL CHECK FOR SIMULATION MODE RUN 1

Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2$ , m <sup>2</sup> /sec <sup>2</sup> $\Delta D$	$k_D$	Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2$ , m <sup>2</sup> /sec <sup>2</sup> $\Delta D$	$k_D$
0.2	1	-4.712	49.537	-0.669	23.3	34	1.936 x 10 <sup>-4</sup>	1.532 x 10 <sup>-5</sup>	0.050
0.9	3	3.279 x 10 <sup>-3</sup>	1.862 x 10 <sup>-5</sup>	0.760	29.3	40	8.003 x 10 <sup>-4</sup>	1.503 x 10 <sup>-5</sup>	0.206
1.3	5	-1.076 x 10 <sup>-3</sup>	9.215 x 10 <sup>-6</sup>	-0.354	67.9	46	1.965 x 10 <sup>-3</sup>	7.112 x 10 <sup>-5</sup>	0.234
2.2	7	-4.251 x 10 <sup>-3</sup>	9.516 x 10 <sup>-6</sup>	-1.381	122.3	53	1.101 x 10 <sup>-4</sup>	9.366 x 10 <sup>-5</sup>	0.011
2.9	9	7.348 x 10 <sup>-3</sup>	9.279 x 10 <sup>-6</sup>	2.415	159.9	58	-1.665 x 10 <sup>-4</sup>	6.931 x 10 <sup>-5</sup>	-0.020
3.3	11	4.793 x 10 <sup>-4</sup>	9.173 x 10 <sup>-6</sup>	0.158	198.6	63	1.055 x 10 <sup>-4</sup>	6.933 x 10 <sup>-5</sup>	0.013
4.2	13	-2.747 x 10 <sup>-3</sup>	9.351 x 10 <sup>-6</sup>	-0.900	202.3	66	8.509 x 10 <sup>-4</sup>	1.416 x 10 <sup>-5</sup>	0.226
4.9	15	1.217 x 10 <sup>-3</sup>	9.272 x 10 <sup>-6</sup>	0.398	203.3	67	-1.947 x 10 <sup>-5</sup>	1.339 x 10 <sup>-5</sup>	-0.005
6.6	17	3.359 x 10 <sup>-4</sup>	1.995 x 10 <sup>-5</sup>	0.075	204.3	68	-5.979 x 10 <sup>-5</sup>	1.340 x 10 <sup>-5</sup>	-0.016
8.3	19	-1.162 x 10 <sup>-3</sup>	1.515 x 10 <sup>-5</sup>	-0.299	205.3	69	2.139 x 10 <sup>-3</sup>	1.286 x 10 <sup>-4</sup>	-0.188
10.9	21	1.630 x 10 <sup>-3</sup>	1.705 x 10 <sup>-5</sup>	0.395	206.3	70	-8.738 x 10 <sup>-3</sup>	4.975 x 10 <sup>-3</sup>	-0.124
14.3	25	9.199 x 10 <sup>-5</sup>	1.561 x 10 <sup>-5</sup>	0.002	207.3	71	3.537 x 10 <sup>-2</sup>	5.181	0.016
18.6	29	-1.324 x 10 <sup>-3</sup>	1.488 x 10 <sup>-5</sup>	-0.344					

From table 51 one can conclude that the orbit determination process for the first example run is indeed converging. The only residuals outside the  $1\sigma$  a priori values are in the transition region between the Earth and Sun sphere of influence. In fact, by looking at the  $k\sigma$  numbers from 5 days after injection to Martian periapsis passage, it appears that the residual values are smaller than their a priori statistics would suggest. The input for run 1 explains why. Not only is the assumed dynamic noise level higher than the actual dynamic modeling error, but also the actual measurements are ten times as accurate as the measurement accuracy assumed by the estimation algorithm.

Table 52 gives the guidance correction information for the first example run. The table presents the magnitude of each commanded midcourse maneuver, the components of the commanded maneuvers, the components of a perfect correction that would be made in the absence of navigation errors, the actual execution error  $\delta\Delta V$  for each correction, actual errors in target conditions resulting from navigation and execution errors, and the a priori target condition variances coming from the concurrent error analysis. The last two columns are the square root of the a priori dispersions divided into the actual errors.

Recall that the error analysis mode run of a similar problem resulted in an expected value of the first midcourse of 8.638 m/sec. The commanded first correction magnitude for the example run with  $\pm 1/2\sigma$  deviations in all injection components is given by 7.0398 m/sec. The subsequent maneuvers are smaller than the error analysis mode predicted, primarily because the actual execution errors used for example run 1 were small when compared with their a priori statistics.

The success of the guidance algorithm within the simulation mode of STEAP is measured by the final actual trajectory's deviations, in terms of the target conditions, from the nominal. The specified injection errors lead to a B·T error of 26 522 km and a B·R error of 166 170 km. Actual target condition errors after the first midcourse correction are 235.692 and 26.137 km, respectively. Second correction deviations propagated to the target resulted in B·T and B·R errors of 48.899 and 14.488 km. After the final correction, the actual trajectory differs from the original by 1.018 km in B·T and 30.373 km in B·R.



TABLE 52.- GUIDANCE CORRECTION INFORMATION FOR SIMULATION MODE RUN 1

Correction Data	Injection	At 5 days 1st midcourse	At 30 days 2nd midcourse	At 180 days 3rd midcourse
$ \Delta V_{com} $ , m/sec		7.0398	0.0493	0.0975
$\Delta V_{com}$ , m/sec		$\dot{X} = 6.2373$ $\dot{Y} = 0.3993$ $\dot{Z} = 3.2398$	0.0099 -0.0310 0.03690	-0.0620 -0.0693 0.0292
$\Delta V_{perfect}$ , m/sec		$\dot{X} = 6.2512$ $\dot{Y} = 0.3756$ $\dot{Z} = 3.2468$	0.0034 -0.02634 0.0381	-0.0619 -0.0712 0.0422
$\delta \Delta V$ , m/sec		$\dot{X} = 0.0147$ $\dot{Y} = 0.0035$ $\dot{Z} = -0.0387$	-0.0037 0.0152 -0.0131	-0.0077 -0.0075 0.0032
$\Delta B \cdot T_{nav}$ , km		-285.411	67.053	-3.680
$\Delta B \cdot R_{nav}$ , km		283.158	-54.765	24.799
$\Delta B \cdot T_{exec}$ , km		49.718	-115.952	2.662
$\Delta B \cdot R_{exec}$ , km		-212.021	40.317	5.574
$\Delta B \cdot T_{total}$ , km	-26 522	-235.692	-48.899	-1.018
$\Delta B \cdot R_{total}$ , km	166 170	26.137	-14.488	30.373
$\sigma_{B \cdot T}^2$ , km <sup>2</sup>	$2.2526 \times 10^9$	263 442	59 653	15 262
$\sigma_{B \cdot R}^2$ , km <sup>2</sup>	$6.9218 \times 10^{10}$	1 525 522	27 781	30 007
$k_{B \cdot T}$	-.558	-0.458	-0.201	-0.008
$k_{B \cdot R}$	+.632	+0.021	-0.0868	+.175

The first example run of the simulation mode was designed to show that the program was working and that the underlying navigation and guidance process would operate satisfactorily under highly advantageous conditions. As can be seen, the problem was set up specifically to show orbit determination convergence and to demonstrate that the governing guidance policy would bring the actual trajectory close to the nominal at the target.

The input changes for the second example run of the STEAP simulation mode are presented in table 53. The nominal trajectory is only updated at three quasi-linear filtering events instead of five. The actual measurement noise is now the same magnitude as the noise assumed by the estimation algorithm. Most importantly, an actual unmodelled acceleration vector is specified to change the actual trajectory between 5.003 and 7.000 days after injection. Finally, the dynamic noise level has been raised slightly.

The purpose of this second simulation mode run was to test the orbit determination procedure when a completely unmodeled acceleration occurs after the first midcourse correction. This might correspond to some kind of shutoff failure in the midcourse execution engines. The dynamic noise level was increased to give the algorithm a chance to converge. Without any process noise modeled, the algorithm completely fails to pick up the unmodeled acceleration.

Table 54 is a chart of the orbit determination inaccuracies throughout the run. Before the first midcourse correction and the subsequent accelerations of the actual trajectory, the orbit determination inaccuracies are slightly higher than those for the first run because the actual measurements are an order of magnitude less accurate. As would be anticipated, the orbit determination inaccuracies grow very large during and after the unmodeled acceleration -- the algorithm itself has no knowledge of this acceleration and can only use its process noise matrix  $Q$  to downgrade the a priori information. Between the second and third midcourse maneuvers the algorithm significantly reduces the velocity errors and keeps the position errors at a more or less constant level. Finally, during the approach to Mars, the familiar pattern develops; the position errors drop on the incoming hyperbola while the velocity errors increase. Just after closest approach both position and velocity deviations of the estimated trajectory from the actual are very large.

TABLE 53.- CHANGES IN CONDITIONS FOR SIMULATION MODE RUN 2

Quasi-linear filtering events only at 5, 30, and 180 days (after corrections)

Dynamic noise constants all  $10^{-20} \text{ km}^2/\text{sec}^4$

Actual unmodeled acceleration from 5.003 days to 7.000 days

$$\ddot{\delta X} = 1 \times 10^{-8} \text{ km/sec}^2$$

$$\ddot{\delta Y} = -1 \times 10^{-8} \text{ km/sec}^2$$

$$\ddot{\delta Z} = 1 \times 10^{-8} \text{ km/sec}^2$$

Actual measurement accuracy same as assumed

TABLE 54.- ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 2

Trajectory time, days	Meas no.	dX $-\hat{\delta X}$ , km	dY $-\hat{\delta Y}$ , km	dZ $-\hat{\delta Z}$ , km	d $\dot{X}$ $-\hat{\delta \dot{X}}$ , m/sec	d $\dot{Y}$ $-\hat{\delta \dot{Y}}$ , m/sec	d $\dot{Z}$ $-\hat{\delta \dot{Z}}$ , m/sec
0	0	0.5000	-0.5000	0.5000	-1.5000	1.5000	-1.5000
0.2	1	-0.2351	1.7567	-1.0012	-0.0286	0.1059	-0.0696
2.2	7	-0.7359	1.6100	-1.3421	-0.0128	0.0207	-0.0064
3.7	12	0.4409	-6.4503	6.8286	-0.0102	-0.0305	0.0271
5.3	16	7.0934	-16.4382	2.3069	0.1812	-0.3799	0.1699
6.6	17	131.9825	-159.3331	42.5387	1.0307	-1.9658	0.9045
7.9	18	256.5837	-410.7727	174.7988	1.2679	-2.4923	1.2722
8.3	19	213.5021	-378.7605	214.9622	0.9591	-2.1706	1.2034
9.6	20	323.1389	-623.0336	347.2422	0.9601	-2.1781	1.1921
10.9	21	430.7667	-834.2454	438.3642	1.0356	-2.0611	1.1473
12.6	23	366.5367	-851.6230	585.4333	0.6475	-1.6886	1.0615
14.3	25	265.2398	-790.9998	688.7424	0.4175	-1.2738	0.9957
16.9	27	388.6511	-1022.2649	773.2507	0.4966	-1.1904	0.8728
18.6	29	275.5549	-953.0326	840.6043	0.2582	-1.0162	0.8159
20.3	31	177.3887	-884.9866	972.4145	0.1715	-0.8031	0.7937
23.3	34	129.7523	-880.1365	1047.1783	0.1039	-0.6387	0.6951
29.3	40	24.8503	-802.4613	1118.9614	-0.0011	-0.3894	0.5222
44.9	43	9.2582	-1338.1043	1760.0321	0.0103	-0.3878	0.4437
99.3	50	33.1909	-2162.3936	2483.7966	0.1378	-0.0540	0.0451
175.6	60	385.9092	-785.6891	1655.8567	0.0543	0.1038	-0.2068
198.6	63	491.0497	-608.0012	1171.6425	0.0566	0.0825	-0.2491
202.3	66	514.5429	-589.4843	1117.8244	0.0520	0.0751	-0.2528
204.3	68	224.5274	-313.8351	1032.0901	-0.0617	0.1130	-0.2367
206.3	70	265.1279	-285.0069	78.9861	-0.3611	0.3272	-0.3638
207.3	71	-2975.9984	2120.1338	-4919.5444	-49.2897	37.7035	-77.7929

Table 55 presents the actual measurement residuals and the a priori residual statistics for example run 2. Recall that the relatively high dynamic or process noise used for this example is reflected in the comparatively high a priori residual variances. The  $-4.9\sigma$  residual at 3.7 days after injection may be traced to the poor state transition matrices in the Earth-Sun transition region and to the fact that for this second run the actual measurement noise is of the same magnitude as the assumed measurement noise. The measurement made during the unmodeled acceleration interval has a  $54\sigma$  residual, a very unlikely event. However, subsequent measurement residuals, although they are far from consistent with their statistics, do not exhibit the properties associated with a bad case of divergence. When bad divergence occurs, it is usual for each successive residual to have a higher magnitude value for  $k_j$  than the previous one.

In a sense, looking at the residuals, the estimate has moderately diverged from the actual trajectory between the first and second midcourse maneuvers. Since, at the time of the second correction the actual trajectory differs from the original nominal by around 10 000 km in each component, orbit determination inaccuracies of roughly 500 to 1000 km do not constitute a bad case of divergence. After the second correction, due to the influence of the process noise matrix  $Q$  on the a priori residual statistics, the residuals become more consistent with their statistics. They do not become smaller, they just are of a size that is more consistent with the a priori statistics whose additional process noise is based on an unmodeled acceleration occurring throughout the flight. This point is important. Although bad inconsistency between residuals and their a priori statistics is almost always an indication that the orbit determination process is diverging, a lack of statistical inconsistency only indicates that the position and velocity uncertainties are in more or less accord with their a priori covariance matrices. If the covariance numbers themselves are large, as they are for example run 2 with high process noise being added throughout, then the navigation procedure is probably not converging very well. However, when both the measurement residuals are becoming smaller and the residuals remain consistent with their statistics, then the underlying estimation algorithm is almost certainly converging.

TABLE 55.- STATISTICAL CONSISTENCY RESIDUAL CHECK FOR SIMULATION MODE RUN 2

Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2_{\Delta D}$ , m <sup>2</sup> /sec <sup>2</sup>	$k_D$	Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2_{\Delta D}$ , m <sup>2</sup> /sec <sup>2</sup>	$k_D$
0.2	1	-4.7091	49.537	-0.669	20.3	31	6.8428 x 10 <sup>-2</sup>	3.8645 x 10 <sup>-5</sup>	11.01
2.2	7	6.851 x 10 <sup>-4</sup>	3.4311 x 10 <sup>-5</sup>	0.117	23.3	34	5.6795 x 10 <sup>-2</sup>	3.7749 x 10 <sup>-5</sup>	9.26
3.7	12	-2.467 x 10 <sup>-2</sup>	2.5219 x 10 <sup>-5</sup>	-4.925	29.3	40	4.1537 x 10 <sup>-2</sup>	3.6581 x 10 <sup>-5</sup>	6.87
5.3	16	1.4711 x 10 <sup>-1</sup>	2.2597 x 10 <sup>-3</sup>	3.118	44.9	43	-3.8970 x 10 <sup>-2</sup>	4.0599 x 10 <sup>-3</sup>	-0.612
6.6	17	5.5556 x 10 <sup>-1</sup>	1.4654 x 10 <sup>-4</sup>	54.250	99.3	50	-1.1765 x 10 <sup>-1</sup>	5.8997 x 10 <sup>-3</sup>	-1.531
7.9	18	1.8595 x 10 <sup>-1</sup>	1.4768 x 10 <sup>-4</sup>	15.310	175.6	60	6.0639 x 10 <sup>-3</sup>	4.7965 x 10 <sup>-3</sup>	0.088
8.3	19	1.1963 x 10 <sup>-1</sup>	3.7289 x 10 <sup>-5</sup>	1.959	198.6	63	-1.1912 x 10 <sup>-2</sup>	4.7249 x 10 <sup>-3</sup>	-0.174
9.6	20	9.8404 x 10 <sup>-3</sup>	1.4624 x 10 <sup>-4</sup>	0.812	202.3	66	3.4641 x 10 <sup>-3</sup>	1.0403 x 10 <sup>-4</sup>	0.340
10.9	21	-1.1056 x 10 <sup>-1</sup>	1.5197 x 10 <sup>-4</sup>	-8.982	204.3	68	-1.4631 x 10 <sup>-2</sup>	1.0179 x 10 <sup>-4</sup>	-1.454
12.6	23	3.1793 x 10 <sup>-2</sup>	1.4875 x 10 <sup>-4</sup>	2.599	206.3	70	4.9406 x 10 <sup>-2</sup>	3.8547 x 10 <sup>-2</sup>	0.252
14.3	25	1.1907 x 10 <sup>-1</sup>	4.0998 x 10 <sup>-5</sup>	18.620	207.3	71	6.1569	2.9147 x 10 <sup>-2</sup>	0.036
16.9	27	-1.0550 x 10 <sup>-1</sup>	1.5483 x 10 <sup>-4</sup>	-8.480					
18.6	29	5.3564 x 10 <sup>-2</sup>	1.5112 x 10 <sup>-4</sup>	2.294					

Table 56 contains the guidance correction information for simulation mode run 2. Because of the unmodeled acceleration, the second and third midcourse maneuver magnitudes are considerably larger. The second midcourse requires 3.349 m/sec of velocity change and the third midcourse uses 2.537 m/sec. After the third midcourse correction, the actual trajectory still misses the nominal target conditions by 783 km in B·T and 933 km in B·R. The a priori error variances for  $\sigma_{B·T}^2$  and  $\sigma_{B·R}^2$  given at the bottom of table 56 reflect the continual addition of the dynamic noise matrix  $Q$  throughout the trajectory.

The third example run (table 57) of the simulation mode was intended to study the effect of bad a priori statistics on the underlying navigation and guidance process. No dynamic noise was included in the estimation algorithm and all the actual errors were large when compared to their assumed covariance matrices. The actual measurement white noise was also sampled from distributions having standard deviations five times as great as those assumed within the estimation algorithm. The results indicate that either this one particular run was lucky or the nominal dynamics being used have a stabilizing influence on the navigation process.

The initial deviations, as shown in table 58, were 3 km in all the position components and 9 m/sec in all the velocity components. These injection errors correspond to a  $\pm 3\sigma$  error in each component based on the injection covariance matrix. As can be seen from the table, the bad initial deviations plus the relatively bad measurements, the poor state transition matrices, and the possible violation of the linearity assumptions led to a significant orbit determination inaccuracy at the time of the first correction. However, during the long heliocentric phase of the trajectory between the second and final midcourse maneuvers, even with the bad a priori statistics, the algorithm effectively tracked the hypothetical vehicle. The large orbit determination errors after Mars periapsis have been passed always occur and do not represent a sudden divergence.

From the statistical consistency chart (table 59) it is obvious that the measurement residuals were never even moderately consistent with their a priori statistics. The measurement taken 3.7 days after injection, for example, when all the poor modeling focuses together, represents an  $86\sigma$  deviation! However, the  $k_{\Delta D}$  values do not increase throughout as expected, indicating that some enforced stability exists for this problem because of the combined navigation and guidance process.

TABLE 56.- GUIDANCE CORRECTION INFORMATION FOR SIMULATION MODE RUN 2

Correction Data	At 5 days 1st midcourse	At 30 days 2nd midcourse	At 180 days 3rd midcourse
$ \Delta V_{com} $ , m/sec	7.0744	3.3490	2.5372
$\Delta V_{com}$ , m/sec	$\dot{X} = 6.2656$ $\dot{Y} = 0.3480$ $\dot{Z} = 3.2676$	-2.9343 1.4041 -0.7961	1.0838 2.2137 -0.6022
$\Delta V_{perfect}$ , m/sec	$\dot{X} = 6.2512$ $\dot{Y} = 0.3756$ $\dot{Z} = 3.2468$	-2.9549 1.7516 -1.1598	0.8888 2.4681 -1.0745
$\delta \Delta V$ , m/sec	$\dot{X} = 0.0152$ $\dot{Y} = 0.0027$ $\dot{Z} = -0.0388$	0.0040 -0.0034 -0.0141	0.0233 -0.0030 -0.0087
$\Delta B \cdot T_{nav}$ , km	436.750	3195.533	745.091
$\Delta B \cdot R_{nav}$ , km	-182.691	490.629	-933.354
$\Delta B \cdot T_{exec}$ , km	67.835	43.287	37.575
$\Delta B \cdot R_{exec}$ , km	-211.189	-124.331	58.133
$\Delta B \cdot T_{total}$ , km	504.585	3238.816	782.666
$\Delta B \cdot R_{total}$ , km	-393.881	366.298	-932.772
$\sigma^2_{B \cdot T}$ , km <sup>2</sup>	299 299	234 538	1 427 435
$\sigma^2_{B \cdot R}$ , km <sup>2</sup>	1 517 718	388 790	2 531 951
$k_{B \cdot T}$	.922	6.692	.655
$k_{B \cdot R}$	-.320	0.588	-.586

TABLE 57.- CHANGES IN CONDITIONS FOR SIMULATION MODE RUN 3

Quasi-linear filtering events only at 5, 30, and 180 days (after corrections)

No dynamic noise

Actual measurement noise larger than assumed measurement noise

$$\sigma_R^2 = 625 \times 10^{-6} \text{ km}^2 \text{ for all ranges}$$

$$\sigma_R^2 = 225 \times 10^{-12} \text{ km}^2/\text{sec}^2 \text{ for all range rates}$$

Actual initial deviations larger ( $\pm 3\sigma$  in all components)

Actual execution errors larger

	1st correction	2nd correction	3rd correction
k =	-0.02	+0.005	-0.008
S =	6.0 cm/sec	-3.7 cm/sec	15.8 cm/sec
$\delta\alpha =$	$4 \times 10^{-2}$ rad	$-8 \times 10^{-3}$ rad	$2 \times 10^{-2}$ rad
$\delta\beta =$	$-1.5 \times 10^{-2}$ rad	$9 \times 10^{-3}$ rad	$1.6 \times 10^{-2}$ rad



TABLE 58.- ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 3

Trajectory time, days	Meas no.	$dX - \hat{\delta X}$ , km	$dY - \hat{\delta Y}$ , km	$dZ - \hat{\delta Z}$ , km	$d\dot{X} - \hat{\delta \dot{X}}$ , m/sec	$d\dot{Y} - \hat{\delta \dot{Y}}$ , m/sec	$d\dot{Z} - \hat{\delta \dot{Z}}$ , m/sec
0	0	-3.0000	3.0000	-3.0000	9.0000	-9.0000	0.0000
0.2	1	1.5560	-11.1056	5.1791	0.1564	-0.6946	0.3740
0.9	3	-0.6955	-13.9163	10.5464	0.1189	-0.2199	-0.0797
1.3	5	-2.6839	-13.1341	9.1740	0.0975	-0.1820	-0.0707
2.2	7	9.2361	-31.0652	1.8003	0.1586	-0.2405	-0.0878
2.9	9	21.9232	-50.1809	-3.9517	0.1785	-0.2814	-0.0987
3.7	12	502.1509	-504.8696	-276.5530	2.3994	-1.8294	-1.4127
4.9	15	400.8532	-509.7124	-59.6389	2.7121	-2.1139	-1.2482
5.3	16	481.5899	-594.3624	-91.8521	2.2072	-2.5639	-0.8261
6.6	17	271.7178	-519.6791	189.4962	-1.1028	0.0580	1.8802
7.9	18	69.6799	-62.2531	-53.5376	-1.4158	1.8593	0.0681
9.6	20	165.0694	113.4651	-470.0522	-0.6535	1.6147	-1.0003
12.6	23	-53.5210	297.1287	-317.8238	-0.7313	1.2599	-0.3722
16.9	27	-175.0070	439.8286	-288.6674	-0.5904	0.9472	-0.2061
20.3	31	-262.5282	420.3587	-96.5663	-0.5285	0.7247	-0.0155
26.3	37	-176.8813	403.4515	-237.1333	-0.3353	0.5096	-0.0822
44.9	43	-58.0959	711.1331	-794.7269	-0.0042	0.2399	-0.2803
99.3	50	-28.6816	-40.6836	138.3024	0.0033	-0.0610	0.0654
145.3	56	-75.0179	84.8091	226.4378	0.0035	-0.0208	0.0490
175.6	60	-63.5350	2.1806	352.6720	0.0053	-0.0231	0.0398
198.6	63	26.4306	-61.2669	370.9051	0.0531	-0.0340	-0.0003
202.3	66	48.8867	-73.8026	370.2857	0.0542	-0.0366	-0.0020
204.3	68	24.5802	-52.5713	368.6812	0.0329	-0.0283	-0.0047
205.3	69	26.6120	-55.3606	372.1360	0.0160	-0.0393	-0.0089
206.3	70	104.0924	-93.0535	21.7904	-0.0318	0.0540	-0.1069
207.3	71	-1884.8883	1382.6643	-1202.1760	-31.5342	23.0819	-19.9108

TABLE 59.- STATISTICAL CONSISTENCY RESIDUAL CHECK FOR SIMULATION MODE RUN 3

Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2_{\dot{\Delta D}}$ , m <sup>2</sup> /sec <sup>2</sup>	$k_{\dot{\Delta D}}$	Trajectory time, days	Meas No.	$\dot{\Delta D}$ , m/sec	$\sigma^2_{\dot{\Delta D}}$ , m <sup>2</sup> /sec <sup>2</sup>	$k_{\dot{\Delta D}}$
0.2	1	28.2202	49.5366	4.050	20.3	31	-1.4449 x 10 <sup>-2</sup>	1.4456 x 10 <sup>-5</sup>	-3.805
0.9	3	-3.2468 x 10 <sup>-2</sup>	1.8523 x 10 <sup>-5</sup>	-7.552	26.3	37	-5.6932 x 10 <sup>-2</sup>	1.4417 x 10 <sup>-5</sup>	-14.99
1.3	5	-1.9095 x 10 <sup>-2</sup>	9.0713 x 10 <sup>-6</sup>	-6.347	44.9	43	-7.9948 x 10 <sup>-2</sup>	3.5821 x 10 <sup>-5</sup>	-13.36
2.2	7	2.2846 x 10 <sup>-2</sup>	9.0661 x 10 <sup>-6</sup>	7.599	99.3	50	1.5177 x 10 <sup>-2</sup>	3.0822 x 10 <sup>-5</sup>	2.735
2.9	9	1.0092 x 10 <sup>-2</sup>	9.0488 x 10 <sup>-6</sup>	3.360	145.3	56	-3.5468 x 10 <sup>-2</sup>	1.6211 x 10 <sup>-5</sup>	-8.815
3.7	12	2.5967 x 10 <sup>-1</sup>	9.0183 x 10 <sup>-6</sup>	86.44	175.6	60	-2.1111 x 10 <sup>-2</sup>	1.3435 x 10 <sup>-5</sup>	-5.754
4.9	15	7.0800 x 10 <sup>-2</sup>	9.0236 x 10 <sup>-6</sup>	23.58	198.6	63	-2.1202 x 10 <sup>-2</sup>	3.0826 x 10 <sup>-5</sup>	-3.823
5.3	16	5.0828 x 10 <sup>-1</sup>	4.9870 x 10 <sup>-2</sup>	2.276	202.3	66	3.3798 x 10 <sup>-2</sup>	1.2742 x 10 <sup>-5</sup>	9.260
6.6	17	-1.5317 x 10 <sup>-1</sup>	3.8951 x 10 <sup>-5</sup>	-24.58	204.3	68	-1.5446 x 10 <sup>-2</sup>	1.1668 x 10 <sup>-5</sup>	-4.525
7.9	18	-1.1113 x 10 <sup>-1</sup>	1.1493 x 10 <sup>-4</sup>	-10.38	205.3	69	4.2539 x 10 <sup>-4</sup>	2.2159 x 10 <sup>-5</sup>	0.906
9.6	20	3.9018 x 10 <sup>-2</sup>	2.7054 x 10 <sup>-5</sup>	7.51	206.3	70	-4.0985 x 10 <sup>-1</sup>	2.5395 x 10 <sup>-3</sup>	-8.14
12.6	23	-4.0164 x 10 <sup>-2</sup>	1.7095 x 10 <sup>-5</sup>	-9.72	207.3	71	3.3059	1.6130	2.608
16.9	27	-1.4089 x 10 <sup>-3</sup>	1.2513 x 10 <sup>-5</sup>	-0.398					

Table 60 presents the guidance correction information for the third simulation mode run. The navigation inaccuracies at the time of the first midcourse correction cause the commanded  $\Delta V$  of 40 m/sec to be more than 5 m/sec away from the required  $\Delta V$ . The actual impact parameter plane errors after the first midcourse correction are 35 000 and 49 000 km in B·T and B·R. Both the subsequent commanded corrections of 5.39 and 2.62 m/sec are somewhat in error, but despite all the modeling errors, the actual corrected trajectory pierces the B-plane with errors of only 18 km in B·T and 289 km in B·R. In setting up the problem it was thought that, even if the corrected trajectory did reach the Martian sphere of influence, which was considered unlikely, the actual deviations from the nominal target conditions would be two orders of magnitude higher than those that resulted. One possible conclusion, therefore, discounting extreme good luck for this example run, is that for this particular problem the navigation and guidance process is relatively insensitive to bad a priori statistics.

Simulation mode runs 4 and 5 were made with a 15 dimensional augmented state including all nine station location coordinate biases. The details for the runs are presented in table 61. The only difference between the two was that the state transition matrices were computed from analytical patched conic methods for run 4 and by numerical differencing for run 5. Since there was no significant alteration in the guidance correction information for these augmented state runs, the two runs will be compared in depth up to the time of the first midcourse correction.

Tables 62 and 64 give the orbit determination inaccuracies for the two runs over the first five days of the trajectory. For about the first day and a half of the hypothetical flight, the assumption that the trajectory is an Earth-based hyperbola is a fairly accurate one. At 1.3 days after injection, the runs exhibit similar orbit determination inaccuracies. Then, as the Sun begins to exert a non-negligible pull on the spacecraft, the conic state transition matrices no longer represent good approximations. After five days of tracking, the orbit determination accuracies for run 5, which used state transition matrices computed by numerical differencing, are an order of magnitude higher.

TABLE 60.- GUIDANCE CORRECTION INFORMATION FOR SIMULATION MODE RUN 3

Correction Date	At 5 days 1st midcourse	At 30 days 2nd midcourse	At 180 days 3rd midcourse
$ \Delta V_{com} $ , m/sec	40.4853	5.3887	2.6231
$\Delta V_{com}$ , m/sec	X = -32.5252 Y = -4.1989 Z = -20.7245	-4.6855 2.6470 0.2782	2.4977 -0.6717 0.4368
$\Delta V_{perfect}$ , m/sec	X = -37.3665 Y = -2.0863 Z = -19.5097	-4.3559 2.2332 0.3125	2.5241 -0.6404 0.2379
$\delta \Delta V$ , m/sec	X = -0.2087 Y = -0.0550 Z = +0.5658	0.0003 -0.0023 -0.0483	0.0086 0.0076 -0.0387
$\Delta B \cdot T_{nav}$ , km	35 847.544	-4 878.663	19.156
$\Delta B \cdot R_{nav}$ , km	-51 877.041	1 870.005	-353.954
$\Delta B \cdot T_{exec}$ , km	-809.939	27.342	-1.581
$\Delta B \cdot R_{exec}$ , km	2850.638	325.837	64.744
$\Delta B \cdot T_{total}$ , km	35 037.605	-4 851.321	17.575
$\Delta B \cdot R_{total}$ , km	-49 026.404	2 195.842	-289.210
$\sigma^2_{B \cdot T}$ , km <sup>2</sup>	8 507 319	159 831	4619.62
$\sigma^2_{B \cdot R}$ , km <sup>2</sup>	31 778 078	597 355	4346.24
$k_{B \cdot T}$	1.202	-12.140	0.250
$k_{B \cdot R}$	-0.871	2.844	-4.398

TABLE 61.- CHANGES IN CONDITIONS FOR SIMULATION MODE RUNS 4 AND 5

Quasi-linear filtering events only at 5, 30, 180 days (after correction)

Dynamic noise constants  $k = 10^{-22} \text{ km}^2/\text{sec}^4$

Assumed and actual measurement noises  $\sigma_D^2 = 25 \times 10^{-6} \text{ km}^2$

$$\sigma_D^2 = 9 \times 10^{-12} \text{ km}^2/\text{sec}^2$$

Same assumed and actual execution errors as in Run 1

Same measurement schedule as in Run 1

Augmented state for both runs  $IAUG = 6$ , dimension of state vector is 15 including 9 station location biases

Actual station location biases

$\Delta R_1 = -10 \text{ m}$	$\Delta \theta_1 = -1 \times 10^{-6} \text{ rad}$	$\Delta \phi_1 = 1 \times 10^{-6} \text{ rad}$
$\Delta R_2 = 10 \text{ m}$	$\Delta \theta_2 = 1 \times 10^{-6} \text{ rad}$	$\Delta \phi_2 = -1 \times 10^{-6} \text{ rad}$
$\Delta R_3 = -10 \text{ m}$	$\Delta \theta_3 = -1 \times 10^{-6} \text{ rad}$	$\Delta \phi_3 = 1 \times 10^{-6} \text{ rad}$

Injection covariance is diagonal

$$\sigma_P^2 \text{ for all positions} = 1 \text{ km}^2$$

$$\sigma_V^2 \text{ for all velocities} = 9 \text{ m}^2/\text{sec}^2$$

$$\sigma_R^2 \text{ for all radius biases} = 400 \text{ m}^2$$

$$\sigma_\theta^2 \text{ for all latitude biases} = 4 \times 10^{-12} \text{ rad}^2$$

$$\sigma_\phi^2 \text{ for all longitude biases} = 4 \times 10^{-12} \text{ rad}^2$$

For Run 4: state transition matrices from patched conic

For Run 5: state transition matrices from numerical differencing

TABLE 62.- ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 4

Trajectory time, days	Meas no.	$\hat{dX} - \delta X$ , km	$\hat{dY} - \delta Y$ , km	$\hat{dZ} - \delta Z$ , km	$\dot{\hat{dX}} - \delta \dot{X}$ , m/sec	$\dot{\hat{dY}} - \delta \dot{Y}$ , m/sec	$\dot{\hat{dZ}} - \delta \dot{Z}$ , m/sec
0	0	0.5000	-0.5000	0.5000	-1.5000	+1.5000	-1.5000
0.2	1	-0.2740	1.7513	-0.9306	-.0304	.1056	-.0665
0.6	2	2.5070	-1.8860	-2.5431	.0410	-.0230	-.0469
0.9	3	0.2669	1.0502	-2.0904	-.0058	.0220	-.0174
1.0	4	1.0826	0.5290	-3.0181	.0026	.0154	-.0262
1.3	5	-0.2731	0.7671	-0.7813	.0157	.0192	-.0001
1.7	6	-0.9928	1.3693	-0.4949	-.0213	.0214	.0014
2.2	7	-5.1645	5.0365	1.8573	-.0503	.0458	.0249
2.6	8	-5.8646	6.3303	1.6258	-.0451	.0445	.0192
2.9	9	-4.0729	4.1684	1.7031	-.0293	.0306	.0131
3.0	10	-4.2665	4.3099	1.2808	-.0340	.0241	.0107
3.3	11	-16.1114	4.7111	22.2463	-.1061	.0267	.1025
3.7	12	-25.0570	20.6528	16.7574	-.1165	.0605	.0604
4.2	13	-32.9581	37.0943	9.0368	-.1168	.0940	.0412
4.6	14	-5.9091	18.8072	16.5039	-.0309	.0332	-.0383
4.9	15	22.4045	-14.7827	-23.4179	.0587	-.0591	-.0613

Tables 63 and 65 compare the estimated values for the nine augmented parameters from the two runs. It should be mentioned that for both runs the processed covariance matrix for the augmented parameters did not undergo any significant change. In neither run did the estimation algorithm "think" it had substantially reduced the parameter errors after five days of tracking. Recall that the augmented state vector run of the error analysis mode also showed little or no change during the flight in the uncertainties associated with the station location biases.

The estimated parameter values of table 63 demonstrate an interesting phenomenon of augmented orbit determination algorithms. Because of the incorrect state transition matrices, the algorithm encounters very large measurement residuals that it must account for by estimating the state vector. It effectively parcels out the error to all available components of the state based on the assumed uncertainties. Since the process noise used for this example is small compared to the dynamic modeling error induced by the improper state transition matrices, the estimation process, in its attempt to lower the residuals, estimates **inordinately** large values for the station location biases. Notice from table 63 that during the first 1.3 days, during which time the conic state transition matrices are still good approximations, the estimated parameter values are still much less than their actual values. During the Earth-Sun Transition region, however, the measurement residuals become larger than anticipated and the parameter estimates vary wildly. In effect, the orbit determination algorithm is trying to explain the residuals by allocating the difference to all components of its state vector. Thus, after five days the process has a ridiculously high estimate of all the station location biases.

By contrast, considering table 65, the simulation mode run with accurate state transition matrices from numerical differencing estimates the augmented parameters more reasonably. At the end of 4.6 days of tracking all parameter estimates are less in magnitude than the actual biases and six of the estimates even have the correct sign.

An important conclusion should be drawn from this demonstration. When augmenting a state vector to include parameters whose uncertainties are relatively small and do not significantly influence the entire process, only if the dynamic model and computational scheme are nearly perfect can any validity be given to the resulting parameter estimates. The smallest amount of dynamic modeling insufficiency, if unaccounted for in the estimation algorithm, can lead to totally erroneous values for the estimated augmented parameters.

TABLE 63.- ESTIMATED PARAMETER VALUES FOR SIMULATION MODE RUN 4

Trajectory time, days	Meas No.	$\delta R_1$ , m	$\delta \theta_1$ , rad	$\delta \phi_1$ , rad	$\delta R_2$ , m	$\delta \theta_2$ , rad	$\delta \phi_2$ , rad	$\delta R_3$ , m	$\delta \theta_3$ , rad	$\delta \phi_3$ , rad
Actual deviations	----	-10,000	-1.000 x 10 <sup>-6</sup>	1.000 x 10 <sup>-6</sup>	10,000	1.000 x 10 <sup>-6</sup>	-1.000 x 10 <sup>-6</sup>	-10,000	1.000 x 10 <sup>-6</sup>	1.000 x 10 <sup>-6</sup>
0.2	1	4.834 x 10 <sup>-3</sup>	-2.279 x 10 <sup>-9</sup>	3.595 x 10 <sup>-10</sup>	0	0	0	0	0	0
0.6	2	5.440 x 10 <sup>-3</sup>	-3.605 x 10 <sup>-9</sup>	1.160 x 10 <sup>-10</sup>	-3.460 x 10 <sup>-2</sup>	-1.496 x 10 <sup>-9</sup>	5.504 x 10 <sup>-10</sup>	0	0	0
0.9	3	-0.2224	-7.376 x 10 <sup>-9</sup>	-4.743 x 10 <sup>-8</sup>	-0.6119	-2.623 x 10 <sup>-8</sup>	9.526 x 10 <sup>-9</sup>	1.342 x 10 <sup>-2</sup>	2.882 x 10 <sup>-8</sup>	4.221 x 10 <sup>-8</sup>
1.0	4	-7.181 x 10 <sup>-2</sup>	7.559 x 10 <sup>-8</sup>	6.192 x 10 <sup>-9</sup>	3.209	-4.387 x 10 <sup>-8</sup>	1.086 x 10 <sup>-7</sup>	-1.474	-3.418 x 10 <sup>-9</sup>	-1.107 x 10 <sup>-7</sup>
1.3	5	0.1302	6.900 x 10 <sup>-8</sup>	1.516 x 10 <sup>-7</sup>	1.028	-5.113 x 10 <sup>-7</sup>	4.695 x 10 <sup>-7</sup>	-1.955	4.446 x 10 <sup>-7</sup>	-6.130 x 10 <sup>-7</sup>
1.7	6	0.9994	4.497 x 10 <sup>-7</sup>	3.627 x 10 <sup>-7</sup>	14.454	-3.579 x 10 <sup>-7</sup>	2.854 x 10 <sup>-8</sup>	-1.796	3.052 x 10 <sup>-7</sup>	-3.852 x 10 <sup>-7</sup>
2.2	7	12.159	-1.593 x 10 <sup>-6</sup>	1.040 x 10 <sup>-6</sup>	-20.716	7.589 x 10 <sup>-7</sup>	-9.233 x 10 <sup>-7</sup>	-4.609	-9.699 x 10 <sup>-7</sup>	-1.234 x 10 <sup>-7</sup>
2.6	8	8.496	-1.428 x 10 <sup>-6</sup>	6.505 x 10 <sup>-7</sup>	-25.492	2.086 x 10 <sup>-7</sup>	-4.323 x 10 <sup>-7</sup>	-3.238	-3.460 x 10 <sup>-7</sup>	-2.337 x 10 <sup>-7</sup>
2.9	9	13.832	-1.373 x 10 <sup>-6</sup>	1.375 x 10 <sup>-6</sup>	-15.343	5.727 x 10 <sup>-7</sup>	-6.272 x 10 <sup>-7</sup>	-1.144	-5.476 x 10 <sup>-7</sup>	-8.518 x 10 <sup>-7</sup>
3.0	10	13.952	-1.343 x 10 <sup>-6</sup>	1.398 x 10 <sup>-6</sup>	-15.433	5.478 x 10 <sup>-7</sup>	-6.022 x 10 <sup>-7</sup>	8.115 x 10 <sup>-2</sup>	-4.863 x 10 <sup>-7</sup>	-8.990 x 10 <sup>-7</sup>
3.3	11	-30.702	2.497 x 10 <sup>-6</sup>	-6.285 x 10 <sup>-7</sup>	38.936	-6.540 x 10 <sup>-7</sup>	3.436 x 10 <sup>-7</sup>	2.194	1.996 x 10 <sup>-6</sup>	4.469 x 10 <sup>-7</sup>
3.7	12	105.933	2.504 x 10 <sup>-6</sup>	1.557 x 10 <sup>-5</sup>	210.277	-3.822 x 10 <sup>-6</sup>	1.046 x 10 <sup>-6</sup>	-49.883	-2.562 x 10 <sup>-6</sup>	-1.634 x 10 <sup>-5</sup>
4.2	13	230.746	-1.5773 x 10 <sup>-5</sup>	2.2509 x 10 <sup>-5</sup>	-164.083	-5.649 x 10 <sup>-6</sup>	3.409 x 10 <sup>-6</sup>	-68.988	-3.830 x 10 <sup>-6</sup>	-2.596 x 10 <sup>-5</sup>
4.6	14	100.139	-8.180 x 10 <sup>-6</sup>	9.642 x 10 <sup>-6</sup>	-192.683	-1.485 x 10 <sup>-5</sup>	1.324 x 10 <sup>-5</sup>	-36.700	8.296 x 10 <sup>-6</sup>	-2.312 x 10 <sup>-5</sup>
4.9	15	105.080	-5.936 x 10 <sup>-6</sup>	1.138 x 10 <sup>-5</sup>	-72.984	-1.083 x 10 <sup>-5</sup>	9.678 x 10 <sup>-6</sup>	-45.575	5.751 x 10 <sup>-6</sup>	-2.161 x 10 <sup>-5</sup>



TABLE 64. - ORBIT DETERMINATION INACCURACIES FOR SIMULATION MODE RUN 5

Trajectory time, days	Meas no.	$\hat{dX} - \delta X$ , km	$\hat{dY} - \delta Y$ , km	$\hat{dZ} - \delta Z$ , km	$\hat{d\dot{X}} - \delta\dot{X}$ , m/sec	$\hat{d\dot{Y}} - \delta\dot{Y}$ , m/sec	$\hat{d\dot{Z}} - \delta\dot{Z}$ , m/sec
0	0	0.5000	-0.5000	0.5000	-1.5000	1.5000	-1.5000
0.2	1	-0.2857	1.7091	-0.8809	-0.0307	0.1036	-0.0638
0.6	2	1.5636	-1.0172	-1.8442	.0233	-.0086	-0.0365
0.9	3	0.4662	0.4467	-1.7204	.0021	.0121	-.0241
1.0	4	0.0703	0.9278	-1.6032	-.0022	.0165	-.0224
1.3	5	-0.7923	1.2256	-0.4235	-.0107	.0161	-.0040
1.7	6	-1.3282	1.1944	0.4123	-.0115	.0104	.0034
2.2	7	-2.0538	1.8465	0.7323	-.0134	.0118	.0052
2.6	8	-2.6544	2.3282	1.0518	-.0142	.0120	.0059
2.9	9	-1.4185	0.8245	0.7123	-.0056	.0037	.0024
3.0	10	0.2937	-0.9901	-0.1088	.0022	-.0044	-.0024
3.3	11	-0.0277	-1.3591	0.9143	-.0002	-.0042	.0027
3.7	12	-0.1391	-1.1222	0.7025	-.0004	-.0031	.0016
4.2	13	-0.1647	-1.3024	0.8215	-.0005	-.0034	.0017
4.6	14	0.3296	-1.7062	0.3309	.0008	-.0042	.0001

TABLE 65.- ESTIMATED PARAMETER VALUES FOR SIMULATION MODE RUN 5

Trajectory time, days	Meas No.	$\delta R_1$ , m	$\delta \theta_1$ , rad	$\delta \phi_1$ , rad	$\delta R_2$ , m	$\delta \theta_2$ , rad	$\delta \phi_2$ , rad	$\delta R_3$ , m	$\delta \theta_3$ , rad	$\delta \phi_3$ , rad
Actual deviations	----									
0.2	1	-10.000	-1.000 x 10 <sup>-6</sup>	-1.000 x 10 <sup>-6</sup>	10.000	1.000 x 10 <sup>-6</sup>	-1.000 x 10 <sup>-6</sup>	-10.000	1.000 x 10 <sup>-6</sup>	1.000 x 10 <sup>-6</sup>
0.6	2	4.876 x 10 <sup>-3</sup>	-2.292 x 10 <sup>-9</sup>	-2.292 x 10 <sup>-9</sup>	0	0	0	0	0	0
0.9	3	4.374 x 10 <sup>-3</sup>	-2.865 x 10 <sup>-9</sup>	-2.865 x 10 <sup>-9</sup>	-1.732 x 10 <sup>-2</sup>	1.017 x 10 <sup>-10</sup>	2.068 x 10 <sup>-10</sup>	0	0	0
1.3	5	-4.965 x 10 <sup>-1</sup>	-2.052 x 10 <sup>-7</sup>	-2.052 x 10 <sup>-7</sup>	-5.680	-1.579 x 10 <sup>-7</sup>	-1.884 x 10 <sup>-7</sup>	-8.661 x 10 <sup>-1</sup>	-9.889 x 10 <sup>-9</sup>	3.452 x 10 <sup>-7</sup>
2.2	7	-3.938 x 10 <sup>-1</sup>	-2.699 x 10 <sup>-7</sup>	-2.699 x 10 <sup>-7</sup>	-8.863	-1.060 x 10 <sup>-7</sup>	-4.036 x 10 <sup>-8</sup>	-9.246 x 10 <sup>-1</sup>	1.442 x 10 <sup>-7</sup>	1.503 x 10 <sup>-7</sup>
2.9	9	-9.819 x 10 <sup>-3</sup>	-5.393 x 10 <sup>-7</sup>	-5.393 x 10 <sup>-7</sup>	-10.752	-4.993 x 10 <sup>-8</sup>	-1.031 x 10 <sup>-6</sup>	-4.082 x 10 <sup>-1</sup>	-5.244 x 10 <sup>-7</sup>	1.084 x 10 <sup>-6</sup>
3.0	10	3.236	-5.075 x 10 <sup>-7</sup>	-5.075 x 10 <sup>-7</sup>	-4.881	3.836 x 10 <sup>-7</sup>	-1.147 x 10 <sup>-6</sup>	8.763 x 10 <sup>-1</sup>	-6.628 x 10 <sup>-7</sup>	7.784 x 10 <sup>-7</sup>
3.3	11	5.266	-5.093 x 10 <sup>-7</sup>	-5.093 x 10 <sup>-7</sup>	-4.735	6.226 x 10 <sup>-7</sup>	-4.820 x 10 <sup>-7</sup>	-1.389	-1.547 x 10 <sup>-7</sup>	-1.327 x 10 <sup>-7</sup>
3.7	12	-6.753 x 10 <sup>-1</sup>	-1.359 x 10 <sup>-7</sup>	-1.359 x 10 <sup>-7</sup>	-1.752	3.617 x 10 <sup>-7</sup>	-3.975 x 10 <sup>-7</sup>	-6.918 x 10 <sup>-1</sup>	1.279 x 10 <sup>-7</sup>	4.975 x 10 <sup>-9</sup>
4.2	13	2.871 x 10 <sup>-1</sup>	-2.843 x 10 <sup>-7</sup>	-2.843 x 10 <sup>-7</sup>	-4.823	4.289 x 10 <sup>-7</sup>	-1.590 x 10 <sup>-7</sup>	-1.521	9.261 x 10 <sup>-9</sup>	-2.617 x 10 <sup>-7</sup>
4.6	14	-3.053 x 10 <sup>-1</sup>	-1.982 x 10 <sup>-7</sup>	-1.982 x 10 <sup>-7</sup>	-3.053	3.983 x 10 <sup>-7</sup>	-1.719 x 10 <sup>-7</sup>	-1.442	9.605 x 10 <sup>-9</sup>	-2.186 x 10 <sup>-7</sup>
		-8.441 x 10 <sup>-1</sup>	-2.209 x 10 <sup>-7</sup>	-2.209 x 10 <sup>-7</sup>	-6.756	3.239 x 10 <sup>-7</sup>	-8.812 x 10 <sup>-8</sup>	-8.574 x 10 <sup>-1</sup>	3.711 x 10 <sup>-7</sup>	-2.304 x 10 <sup>-7</sup>

Table 66 presents numerous elements, at selected trajectory epochs, of the state transition matrices used for runs 4 and 5. The larger terms are in significant agreement throughout, although two of the smaller components,  $\phi_{15}$  and  $\phi_{24}$ , disagree considerably in the transition region. The relatively small differences displayed in the tables is all that is necessary for the improper parameter estimation of run 4.

TABLE 66.- STATE TRANSITION MATRIX COMPARISON FOR SIMULATION MODE RUNS 4 AND 5

Beginning trajectory time, days	Beginning trajectory time, days	$\Phi_{11}$ CONIC	$\Phi_{11}$ NDIF	$\Phi_{22}$ CONIC	$\Phi_{22}$ NDIF	$\Phi_{66}$ CONIC	$\Phi_{66}$ NDIF	$\Phi_{14}$ CONIC	$\Phi_{14}$ NDIF
0.0	0.0	-3.8405	-3.8252	12.5624	12.5499	0.99131	0.99137	$2.6722 \times 10^4$	$2.6700 \times 10^4$
0.2	0.6	1.0853	1.0852	0.95759	0.95790	0.98430	0.98410	$3.5103 \times 10^4$	$3.5101 \times 10^4$
0.6	0.9	1.0044	1.0042	0.99840	0.99862	0.99806	0.99802	$2.5951 \times 10^4$	$2.5950 \times 10^4$
1.0	1.3	1.0012	1.0012	0.99964	0.99962	0.99934	0.99933	$2.5929 \times 10^4$	$2.5929 \times 10^4$
1.7	2.2	1.0007	1.0008	0.99981	0.99982	0.99956	0.99952	$4.3209 \times 10^4$	$4.3210 \times 10^4$
2.2	2.6	1.0002	1.0003	0.99994	0.99996	0.99984	0.99981	$3.4563 \times 10^4$	$3.4563 \times 10^4$
2.6	2.9	1.0001	1.0001	0.99998	0.99999	0.99994	0.99992	$2.5921 \times 10^4$	$2.5921 \times 10^4$
2.9	3.0	1.0000	1.0000	1.0000	1.0000	1.0000	0.99999	$8.6400 \times 10^3$	$8.6400 \times 10^3$
3.0	3.3	1.0000	1.0001	1.0000	1.0000	0.99999	0.99994	$2.5920 \times 10^4$	$2.5921 \times 10^4$
3.3	3.7	1.0000	1.0001	1.0000	1.0000	0.99998	0.99992	$3.4560 \times 10^4$	$3.4561 \times 10^4$
3.7	4.0	1.0000	1.0000	1.0000	1.0000	0.99999	0.99996	$2.5920 \times 10^4$	$2.5920 \times 10^4$
4.2	4.6	1.0000	1.0000	1.0000	1.0000	0.99998	0.99995	$3.4560 \times 10^4$	$3.4560 \times 10^4$
3.0	5.0	1.0000	1.0019	1.0006	1.0002	0.99943	0.99852	$1.7280 \times 10^5$	$1.7288 \times 10^5$
0.0	0.2	$1.4846 \times 10^4$	$1.4847 \times 10^4$	$1.6300 \times 10^4$	$1.6301 \times 10^4$	$-1.2725 \times 10^4$	$-1.2736 \times 10^4$	$-3.7439 \times 10^3$	$-3.7429 \times 10^3$
0.2	0.6	$3.4304 \times 10^4$	$3.4309 \times 10^4$	$3.4294 \times 10^4$	$3.4292 \times 10^4$	$6.6601 \times 10^2$	$6.6711 \times 10^2$	$6.6696 \times 10^2$	$6.6807 \times 10^2$
0.6	0.9	$2.5909 \times 10^4$	$2.5911 \times 10^4$	$2.5900 \times 10^4$	$2.5900 \times 10^4$	$4.7276 \times 10^1$	$4.6129 \times 10^1$	$4.7277 \times 10^1$	$4.6133 \times 10^1$
1.0	1.3	$2.5917 \times 10^4$	$2.5917 \times 10^4$	$2.5914 \times 10^4$	$2.5914 \times 10^4$	$1.5014 \times 10^1$	$1.4346 \times 10^1$	$1.5014 \times 10^1$	$1.4327 \times 10^1$
1.7	2.2	$4.3198 \times 10^4$	$4.3198 \times 10^4$	$4.3193 \times 10^4$	$4.3192 \times 10^4$	$1.6579 \times 10^1$	$1.5947 \times 10^1$	$1.6579 \times 10^1$	$1.5887 \times 10^1$
2.2	2.6	$3.4559 \times 10^4$	$3.4560 \times 10^4$	$3.4558 \times 10^4$	$3.4558 \times 10^4$	$4.7318$	$4.4260$	$4.7318$	$4.3652$
2.6	2.9	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$2.5919 \times 10^4$	$2.5919 \times 10^4$	$1.3576$	$1.2302$	$1.3576$	$1.2059$
2.9	3.0	$8.6400 \times 10^3$	$8.6400 \times 10^3$	$8.6400 \times 10^3$	$8.6400 \times 10^3$	$-7.166 \times 10^{-3}$	$3.8147 \times 10^{-2}$	$-7.172 \times 10^{-3}$	$2.4796 \times 10^{-2}$
3.0	3.3	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$2.5919 \times 10^4$	$-1.5468 \times 10^{-1}$	$7.6771 \times 10^{-1}$	$-1.5468 \times 10^{-1}$	$7.9584 \times 10^{-1}$
3.3	3.7	$3.4560 \times 10^4$	$3.4560 \times 10^4$	$3.4560 \times 10^4$	$3.4559 \times 10^4$	$-3.6747 \times 10^{-1}$	$1.2865$	$-3.6747 \times 10^{-1}$	$1.3046$
3.7	4.0	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$2.5920 \times 10^4$	$-1.5532 \times 10^{-1}$	$3.7622 \times 10^{-1}$	$-1.5531 \times 10^{-1}$	$3.3474 \times 10^{-1}$
4.2	4.6	$3.4560 \times 10^4$	$3.4560 \times 10^4$	$3.4560 \times 10^4$	$3.4559 \times 10^4$	$-3.7186 \times 10^{-1}$	$4.9257 \times 10^{-1}$	$-3.7186 \times 10^{-1}$	$4.8971 \times 10^{-1}$
3.0	5.0	$1.7283 \times 10^5$	$1.7281 \times 10^5$	$1.7277 \times 10^5$	$1.7270 \times 10^5$	$-4.2673 \times 10^1$	$1.0511 \times 10^2$	$-4.2673 \times 10^1$	$1.0448 \times 10^2$

Martin Marietta Corporation  
 Denver, Colorado  
 August 15, 1969

## VIII. REFERENCES

1. Novak, D. H.: Virtual Mass Technique for Computing Space Trajectories. ER-14045, Martin Marietta Corporation, Baltimore, Maryland, Jan. 1966 (Completed under Contract NAS9-4370).
2. Jazwinski, A. H.: Optimal Trajectories and Linear Control of Nonlinear Systems. AIAA J., -2, no. 8, Aug. 1964.
3. Lewallen, J. M.; Papley, B. D.; and Williams, S. D.: Iterative Procedures for Indirect Trajectory Optimization Methods. J. Spacecraft Rockets, 5, no. 3, Mar. 1968.
4. McGill, R.; and Kenneth, P.: Solution of Variational Problems by Means of a Generalized Newton - Raphson Operator. AIAA J., -2, no. 10, Oct. 1964.
5. Bellman, R.; Kalaba, P.; and Wing, G. M.: Invariant Imbedding and the Reduction of Two Point Boundary Value Problems to Initial Value Problems. Proc. Natl. Acad. Sci. U. S., 46, no. 12, 1960.
6. Danby, J. M. A.: Matrizant of Keplerian Motion. AIAA J., -3, no. 4, Apr. 1965.
7. Kizner, W.: A Method of Describing Miss Distances for Lunar and Interplanetary Trajectories under the Influences of Multiple Planet Attractions. TR-32-464, Jet Propulsion Laboratory, 1963.
8. Joseph, A. E.; and Richard, R. J.: Space Research Conic Program. Engineering Planning Document 406, Jet Propulsion Laboratory, July 1966.
9. Battin, R. H.: Astronautical Guidance. McGraw-Hill Book Company, Inc, New York, 1964.
10. Lee, Gentry; Falce, Ralph; and Hopper, Fred: Interplanetary Trajectory Error Analysis, Volume I - Analytical Manual, Volume II - Computer Program Design. MCR-67-441, Martin Marietta Corporation, Denver, Colorado, Dec. 1967 (Completed under Contract NAS8-21120).

11. Danby, J. M. A.: The Matrizant of Keplerian Motion. AIAA J., vol. 2, no. 1, Jan. 1964.
12. Hoffman, L. and Young, G.: Private communication, NASA-Langley working paper, in-house document.
13. Jazwinski, A. H.: Adaptive Filtering. Analytical Mechanics Associates, Lanham, Maryland, May 1968.